

**UNIVERSIDAD AUTÓNOMA DE ICA**  
**FACULTAD DE CIENCIAS, INGENIERÍA Y ADMINISTRACIÓN**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**“SISTEMATIZACIÓN DE PROCESOS PARA LA GESTIÓN  
DE LA INFORMACIÓN DOCUMENTARIA ENTRE LAS  
ÁREAS DE VENTAS, ADMINISTRACIÓN Y  
CONSTRUCCIÓN DE LA EMPRESA CONSTRUCTORA  
AIF CONSTRATISTAS GENERALES S.A.C”  
CHINCHA ALTA – 2015**

**TESIS**

Presentada para optar el Título Profesional de Ingeniero de  
Sistemas

**PRESENTADA POR:**

Inca García, Rubén Alejandro  
Vera Orellana, Brendalíz Geraldine

CHINCHA - PERÚ – 06-03-2015

“SISTEMATIZACIÓN DE PROCESOS PARA LA GESTIÓN DE LA  
INFORMACIÓN DOCUMENTARIA ENTRE LAS ÁREAS DE VENTAS,  
ADMINISTRACIÓN Y CONSTRUCCIÓN DE LA EMPRESA  
CONSTRUCTORA  
"AIF CONSTRATISTAS GENERALES S.A.C"  
CHINCHA ALTA – 2015

---

AUTOR

Inca García, Rubén Alejandro

---

AUTOR

Vera Orellana, Brendaliz Geraldine

---

ASESOR

Mg Sc. Moreno Heredia, Armando

Presentada a la Facultad de Ciencias, Ingeniería y administración de la  
Universidad Privada Autónoma de Ica. Para optar el Título de  
Ingeniero de Sistemas

APROBADO POR:

---

PRESIDENTE DEL JURADO

---

SECRETARIO DEL JURADO

---

VOCAL DEL JURADO

Marzo, 2015

## **DEDICATORIA:**

Dedico esta tesis

A Dios por permitirme el haber llegado hasta este momento tan importante en mi formación profesional.

A mis padres quienes me dieron vida, educación, apoyo, valores y consejos, que permitieron que siempre este por el sendero del bien.

A mi hermana mayor por el ejemplo y apoyo que me dio desde pequeño.

Y mi hermana menor por ser mi compañía.

Para todos ellos hago esta dedicatoria.

Inca García, Rubén Alejandro

## **DEDICATORIA:**

Dedico este proyecto de tesis a Dios verdadera fuente de amor y sabiduría que ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar. A mi padre, porque gracias a él sé que la responsabilidad se la debe vivir como un compromiso de dedicación y esfuerzo. A mi madre, cuyo vivir me ha mostrado que en el camino hacia la meta se necesita fortaleza para aceptar las derrotas y del sutil coraje para derribar miedos. Quienes a lo largo de mi vida han velado por mi bienestar siendo mi apoyo en todo momento. Depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad. Es por ello que soy lo que soy ahora. Los amo con mi vida.

Vera Orellana, Brendaliz Geraldine

## **AGRADECIMIENTO:**

Nuestro eterno agradecimiento a Dios, por llenar nuestra vida de bendiciones, por estar presente en cada paso, y permitir poco a poco lograr los objetivos propuestos y por haber puesto en nuestro camino a grandes personas, las cuales nos brindaron su apoyo constante e incondicional.

A nuestros docentes que nos enseñaron en nuestra etapa universitaria, en especial a nuestros docentes Mg Sc. Moreno Heredia, Armando y al Ing. Barrios Quispe, Richard que nos aconsejaron en el desarrollo de nuestra tesis.

A todas las personas de la empresa AIF Contratistas Generales SAC que nos brindaron su apoyo e información desinteresadamente para la realización de esta tesis.

Por último mil gracias a nuestros padres y hermanos por siempre estar a nuestro lado, por haber sido una parte muy importante de esta etapa de nuestra vida, apoyándonos siempre de forma incondicional y constante.

Inca García, Rubén Alejandro  
Vera Orellana, Brendaliz Geraldine

## INDICE

	<b>PAG</b>
RESUMEN.....	X
ABTRACT.....	xi
INTRODUCCIÓN.....	1
1. CAPITULO I: ANALISIS DEL OBJE TI DE ESTUDIO Y ASPECTOS	
DE LA INVESTIGACIÓN .....	3
1.1. DATOS GENERALES DE LA EMPRESA.....	4
1.1.1. RAZÓN SOCIAL.....	4
1.1.2. REPRESENTANTE LEGAL.....	4
1.1.3. RUBRO DE LA EMPRESA.....	4
1.1.4. AÑO DE INICIO DE ACTIVIDADES EN EL SECTOR.....	4
1.1.5. UBICACIÓN GEOGRÁFICA.....	4
1.1.6. PLANO DE UBICACIÓN.....	5
1.1.7. ACTIVIDADES PRINCIPALES.....	5
1.1.8. ORGANIGRAMAS.....	6
1.1.9. HARDWARE Y SOFTWARE.....	9
1.2. SITUACIÓN PROBLEMÁTICA.....	10
1.3. PROBLEMA.....	11
1.4. HIPÓTESIS.....	11
1.5. OBJETIVOS.....	11
1.5.1. OBJETIVO GENERAL.....	11
1.5.2. OBJETICOS ESPECIFICOS.....	12
1.6. JUSTIFICACIÓN E IMPORTANCIA.....	12
1.7. METODOLOGÍA.....	13
1.7.1. MODO DE INVESTIGACIÓN.....	13
1.7.2. TIPO DE INVESTIGACIÓN.....	13
1.7.3. CONTRASTACIÓN DE HIPÓTESIS.....	13
1.7.4. POBLACION Y MUESTRA ESTADISTICA DE ESTUDIO.....	14
1.7.5. TECNICA E INSTRUMENTO DE RECOPIACION	
DE DATOS.....	16
2. CAPITULO II: MARCO TEÓRICO.....	17
2.1. REVISION BIBLIOGRÁFICA.....	18

2.2. BASE TEÓRICA.....	24
2.2.1. LAS METODOLOGIAS ÁGILES.....	24
2.2.1.1. XP – EXTREME PROGRAMMING.....	28
2.2.1.2. SCRUM.....	33
2.2.1.3. RAD.....	40
2.2.2. MICROSOFT .NET.....	43
2.2.2.1. VENTAJAS .NET.....	44
2.2.2.2. COMPONENTES FUNDAMENTALES.....	45
2.2.2.2.1. NET FRAMEWORK.....	45
2.2.2.2.2. CLR-COMMON LANGUAGE RUNTIMME.....	45
2.2.2.2.3. CLASS LIBRARY.....	46
2.2.2.2.4. CLS-COMMON LANGUAGE SPECIFICATION..	48
2.2.2.3. BIBLIOTECAS PRINCIPALES.....	48
2.2.2.3.1. BASE CLASS LIBRARY.....	48
2.2.2.3.2. ADO .NET.....	49
2.2.2.3.3. ASP .NET.....	52
2.2.2.4. VISUAL STUDIO 2010.....	54
2.2.2.4.1. LEGUAJES.....	56
2.2.2.4.1.1. LENGUAJE VISUAL AL BASIC.....	57
2.2.2.4.1.2. LENGUAJE C#.....	57
2.2.2.4.1.3. VISUAL F#.....	57
2.2.2.4.2. ACCESO A DATOS Y MODELADO.....	58
2.2.2.4.3. PROGRAMACIÓN PARALELA.....	58
2.2.2.5. MODELADOR DE BASE DE DATOS ERWIN.....	59
2.2.2.6. BASE DE DATOS.....	59
2.2.2.7. MICROSOFT SQL SERVER 2008.....	60
2.2.2.7.1. EDICIONES.....	61
2.2.2.8. WINDOWS 7.....	65
2.2.2.9. WINDOWS SERVER 2008.....	66
2.2.2.10. HERRAMIENTAS ADICIONALES PARA EL DESARROLLO.....	67
2.2.2.10.1. JQUERY.....	67
2.2.2.10.2. GANTTPROJECT.....	69
2.2.2.11. TECNOLOGÍAS ADICIONALES PARA EL DESARROLLO.....	72

2.2.2.11.1.	HTML5.....	72
2.2.2.11.2.	CSS3.....	72
2.2.2.11.3.	AJAX.....	75
2.2.2.12.	RED DE DATOS.....	78
2.2.2.12.1.	CLASES DE REDES DE DATOS.....	78
2.2.2.12.2.	TOPOLOGÍAS.....	79
2.2.2.12.3.	RED E INTERNET DE AIF.....	82
2.2.2.12.3.1.	RED.....	82
2.2.2.12.3.2.	INTERNET.....	85
3.	CAPITULO III: PLANIFICACION DEL PROYECTO.....	86
3.1.	PLANIFICACIÓN DEL PROYECTO.....	87
3.1.1.	ALCANCE DEL SISTEMA.....	87
3.1.2.	CRONOGRAMA DE ACTIVIDADES.....	88
3.1.3.	FACTORES CRITICOS.....	89
3.1.4.	SUPUESTOS.....	89
3.1.5.	RESTRICCIONES.....	89
3.1.6.	ANALISIS DE FACTIBILIDAD.....	90
4.	CAPITULO IV: DESARROLLO DEL SISTEMA.....	93
4.1.	ANÁLISIS DEL SISTEMA.....	94
4.1.1.	DEFINICION DE ESTÁNDARES.....	94
4.2.	HISTORIAS DE USUARIO.....	97
4.3.	PLAN DE ENTREGAS.....	104
4.4.	DISEÑO DEL SISTEMA.....	106
4.4.1.	DEFINICION DE ESTÁNDARES.....	106
4.5.	DISEÑO DE LA BASE DE DATOS.....	108
4.6.	CONSTRUCCION DEL SISTEMA POR SPRINT.....	111
4.6.1.	SPRINT 1.....	111
4.6.2.	SPRINT 2.....	111
4.6.3.	SPRINT 3.....	112
4.6.4.	SPRINT 4.....	113
4.6.5.	SPRINT 5.....	114
4.6.6.	SPRINT 6.....	115
4.6.7.	SPRINT 7.....	116
4.6.8.	SPRINT 8.....	116
5.	CONCLUSIONES.....	117

6. RECOMENDACIONES.....	119
7. REFERENCIAS BIBLIOGRÁFICAS.....	121
7.1. BIBLIOGRAFÍA.....	122
7.2. LINKOGRAFÍA.....	123
8. ANEXOS.....	124
8.1. ANEXO N° 1: MATRIZ DE CONSISTENCIA.....	125
8.2. ANEXO N° 2: MANUAL DE USUARIO.....	126
8.3. ANEXO N° 3: MANUAL DEL PROGRAMADOR.....	140

## ÍNDICE DE ILUSTRACIONES

Ilustración 1: Ubicación de Chincha .....	4
Ilustración 2: Ubicación de AIF .....	5
Ilustración 3: Organigrama estructural. ....	6
Ilustración 4 Organigrama nominal.....	7
Ilustración 5: Organigrama funcional .....	8
Ilustración 6: Plataforma Microsoft .net .....	44
Ilustración 7: Namespace SYSTEM .....	47
Ilustración 8: Principales Namespace de BCL .....	49
Ilustración 9: Arquitectura ADO .NET.....	49
Ilustración 10: Estructura de Red primer piso .....	82
Ilustración 11: Estructura de Red segundo piso .....	83
Ilustración 12: Cronograma de Actividades del Proyecto .....	88
Ilustración 13: Ejemplo de Pila de producto .....	96
Ilustración 14: Diseño BD part1 .....	108
Ilustración 15: Diseño BD part2 .....	109
Ilustración 16: Diseño BD part3 .....	110
Ilustración 17: Registrar Captado .....	111
Ilustración 18: Registrar beneficiario .....	111
Ilustración 19: Reporte captados .....	112
Ilustración 20: Reporte de beneficiarios .....	112
Ilustración 21: Actualizar beneficiario pre FMV .....	113
Ilustración 22: Actualizar código FMV .....	113
Ilustración 23: Ingreso fecha de ahorro .....	114
Ilustración 24: Asignación carta fianza .....	114
Ilustración 25: Actualizar estado de obra .....	115
Ilustración 26: Cuadro de ejecución .....	115
Ilustración 27: Estructura de Red primer piso .....	116

## INDICE DE TABLAS

Tabla 1: Hardware.....	9
Tabla 2 : Software .....	10
Tabla 3: Definición de la variable independiente .....	13
Tabla 4: Definición de variable dependiente .....	14
Tabla 5: Diferencias entre Metodologías Agiles y no Agiles.....	27
Tabla 6: Los roles de Scrum.....	38
Tabla 7: Factibilidad técnica software .....	90
Tabla 8: Factibilidad técnica hardware.....	90
Tabla 9: Factibilidad económica software .....	91
Tabla 10: Factibilidad económica hardware.....	91
Tabla 11: Factibilidad económica Recursos humanos.....	92
Tabla 12: Factibilidad económica Flujo de pago.....	92
Tabla 13: Historia de Usuario Fabiola Vasquez.....	97
Tabla 14: Historia de Usuario Magaly Siguas .....	98
Tabla 15: Historia de Usuario Fiorella Perez.....	99
Tabla 16: Historia de Usuario Juan Carlos de la Cruz .....	100
Tabla 17: Historia de Usuario Roberto Soto.....	101
Tabla 18: Historia de Usuario Ing. Marco Herrera .....	102
Tabla 19: Plan de historias de usuario .....	104

## RESUMEN

La presente investigación de proyecto tiene como objetivo principal la sistematización de procesos para la gestión de la información documentaria entre las áreas de ventas, administración y construcción de la empresa constructora "AIF CONTRATISTAS GENERALES S.A.C" , por lo que se planteó el desarrollo de una aplicación de tramites documentarios, el cual además permita minimizar el tiempo de obtención de documentos, minimizar el tiempo de armado a cada uno de los trámites que deben ser presentados a las municipalidades, minimizar el tiempo de búsqueda de un estado de beneficiario, generar una comunicación más eficiente, confiable y segura en el manejo de documentos en la entidad generando reportes a tiempo, para que la alta dirección pueda tomar decisiones adecuadas.

El desarrollo de este proyecto se realizó mediante la metodología Scrum y XP, una metodología ágil de desarrollo, que tiene como característica principal, permitir al equipo de desarrollo una rápida respuesta al cambio, y así una planificación flexible y abierta. Además de permitir satisfacer al cliente mediante entregas tempranas y continuas de software. Además se utilizaron las siguientes herramientas para el desarrollo: ASP.NET, Microsoft Visual Studio 2010 y Microsoft SQL Server 2008.

El proyecto está conformado por módulos los cuales son: Ventas, el cual se encarga de registrar los captados luego convertirlos en beneficiarios y la recolección de sus respectivas documentaciones; Administración, el cual se encarga de procesar la información actualizando los expedientes y Construcción encargado de la generación de documentación.

## ABSTRACT

This research project's main objective systematization of processes to centralize the documentary information between the areas of sales, management and construction of the building company "AIF Contradicts GENERAL SAC", so that the development of an application of procedures raised documentary, which also minimizes the time of obtaining documents, minimizing assembly time to each of the paperwork that must be submitted to the municipalities, to minimize the search time of a beneficiary status, generate more efficient communication, reliable and secure document management in the state generate reports on time so that senior management can make appropriate decisions.

The development of this project was performed using Scrum and XP methodologies, agile development methodology, whose main feature, allowing the development team a quick response to change, and thus a flexible and open planning. Besides allowing satisfying the customer through early and continuous delivery of software. In addition the following development tools were used: ASP.NET, Microsoft Visual Studio 2010 and Microsoft SQL Server 2008.

The project consists of modules which are: Sales, which is responsible for the register convert then captured in the collection of beneficiaries and their documentation; Directors, which is responsible for processing information and updating records Construction Manager Documentation generation.

## INTRODUCCIÓN

El presente proyecto de investigación lleva por título: "Sistematización de Procesos para la gestión de la Información Documentaria entre las Áreas de Ventas, Administración y Construcción de la Empresa Constructora AIF Contratistas Generales S.A.C", el cual tiene como objetivo principal desarrollar un sistema que permita la automatización y reutilización de la información y los procesos de dicha empresa.

Este proyecto ha sido dividido en siete capítulos: aspecto informativo, planificación del proyecto, desarrollo del proyecto, integración, soporte y mantenimiento, discusión, conclusiones y recomendaciones.

En el primer capítulo, aspecto informativo, se incluyen el marco lógico, el cual presenta la situación problemática, el problema de la investigación, la hipótesis de la misma, la definición de los objetivos, la justificación e importancia y la definición de términos y conceptos utilizados en el desarrollo del proyecto; el marco metodológico, en el cual se plantea el modo y tipo de investigación, la contrastación de la hipótesis, la población y muestra estadística de estudio con la que se trabajará y las técnicas e instrumentos de recopilación de datos a utilizar; y por último el marco teórico, el cual incluye los antecedentes y base teórica de la investigación.

En el segundo capítulo, planificación del proyecto, se incluye el alcance del sistema, el cronograma de actividades del proyecto, los factores críticos de éxito, supuestos, restricciones y análisis de factibilidad.

En el tercer capítulo, desarrollo del sistema, se incluye las fases del desarrollo del mismo, los cuales son análisis, diseño, construcción, pruebas.

En el cuarto capítulo, integración, se incluye a los procesos automatizados, las consideraciones de seguridad y la puesta en marcha.

En el quinto capítulo, soporte y mantenimiento.

En el capítulo sexto, discusión, se presenta un análisis estadístico de los resultados obtenidos en las dos encuestas realizadas al inicio y al final del proyecto con la finalidad de determinar si se lograron los objetivos de la investigación.

Finalmente, en el séptimo capítulo, se incluye las conclusiones y recomendaciones.

## **1. CAPITULO I**

### **ANÁLISIS DEL OBJETO DE ESTUDIO Y ASPECTOS DE LA INVESTIGACIÓN**

## 1.1. DATOS GENERALES DE LA EMPRESA

### 1.1.1. Razón social

AIF CONTRATISTAS GENERALES S.A.C.

Ruc: 20494453003

### 1.1.2. Representante legal

Isaias Blas Latinez Guerovich

### 1.1.3. Rubro de la empresa

Construcción de edificios completos

### 1.1.4. Año de inicio de actividades en el sector

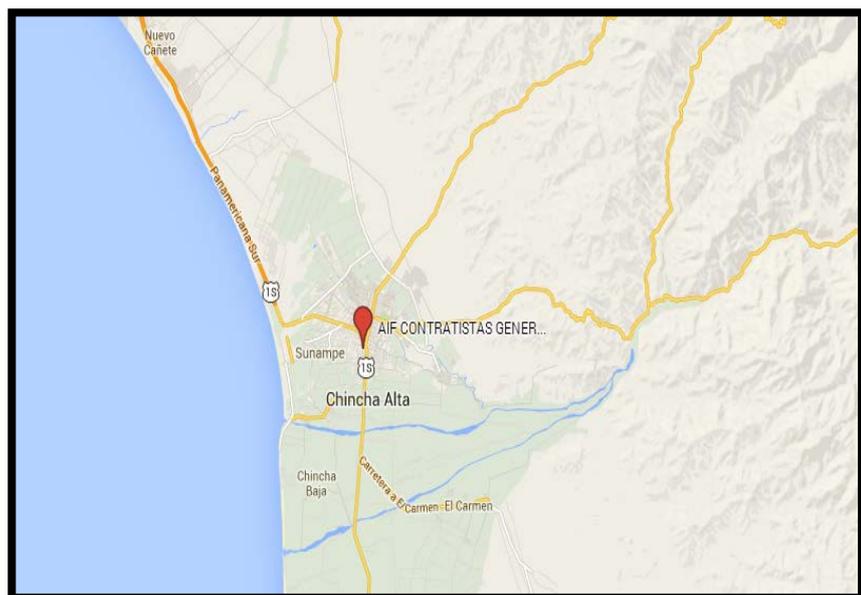
11 de Marzo del 2008

### 1.1.5. Ubicación geográfica

Chincha Alta es una ciudad de la costa sur-central del Perú, capital de la Provincia de Chincha (Departamento de Ica), en la cuenca del río San Juan, a 200 kilómetros al sur de Lima. Tiene una superficie de 238,34 km<sup>2</sup>.

Usualmente es referida imlemente como Chincha.

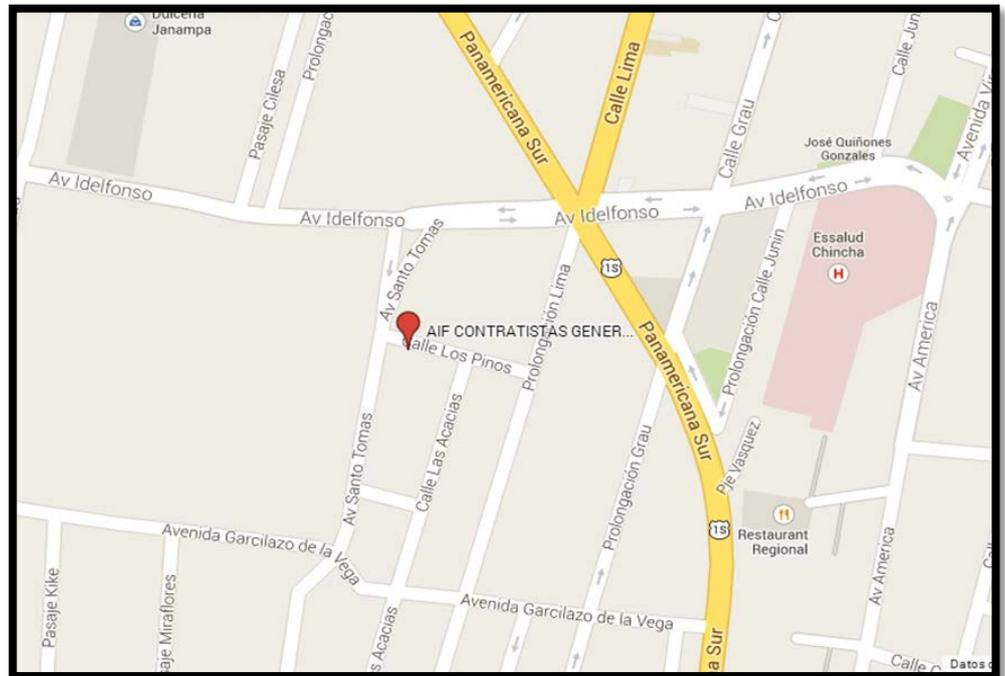
Img.1: Ubicación de Chincha



### 1.1.6. Plano de ubicación

Dirección: urbanización julio arboleda Mz. A Lt. 6- calle los ficus

Img.2: Ubicación de AIF



### 1.1.7. Actividades principales

#### 1.1.7.1. Ventas:

- Captación de beneficiarios.
- Preparación de documentación para el fondo mi vivienda.

#### 1.1.7.2. Administración:

- Seguimiento de los trámites del fondo mi vivienda.
- Realización de asignaciones.
- Desembolsos y liberaciones.

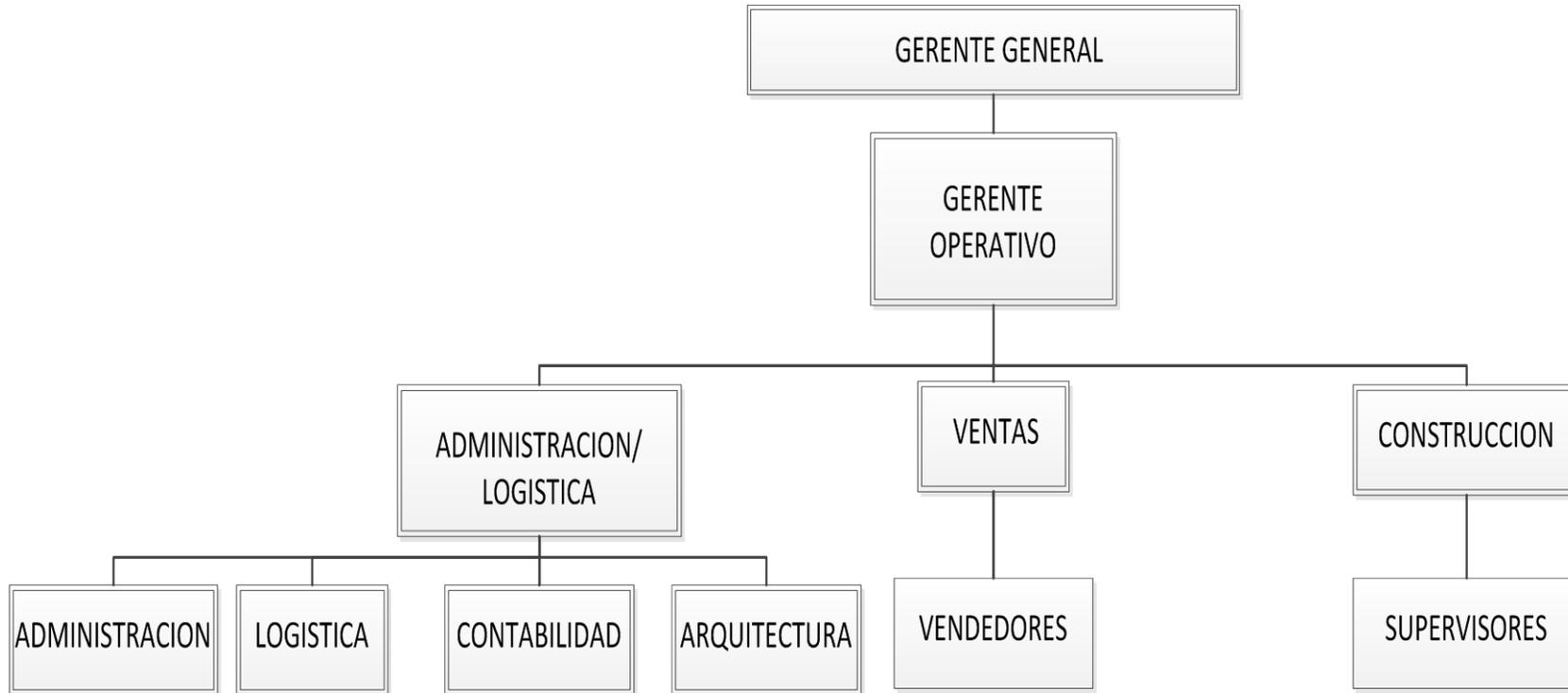
#### 1.1.7.3. Construcción:

- Seguimiento de obras
- Supervisión de calidad en las viviendas construidas.
- Liberación de Cartas Fianzas

### 1.1.8. Organigramas

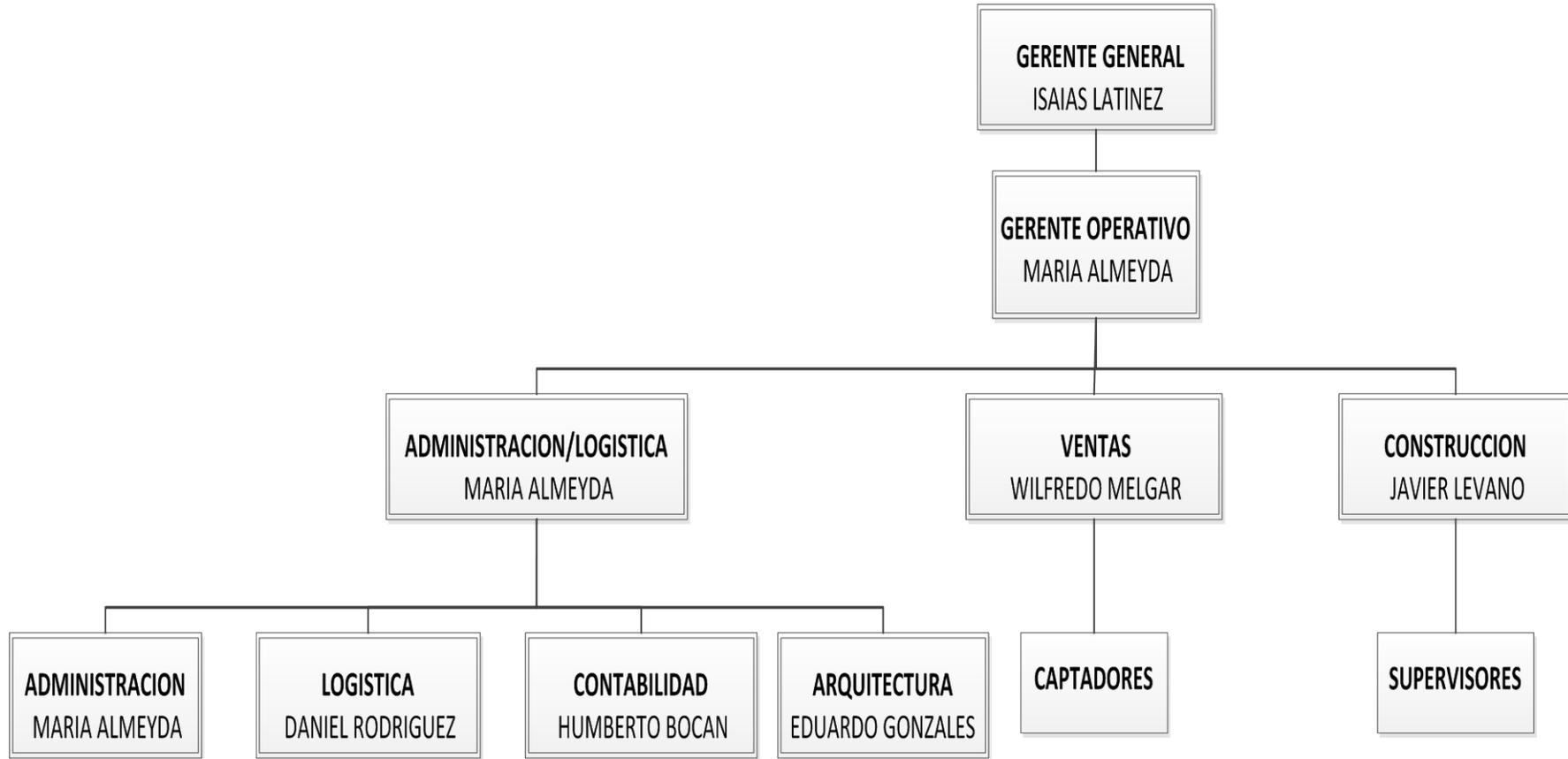
#### 1.1.8.1. Organigrama estructural

Img.3: organigrama estructural.



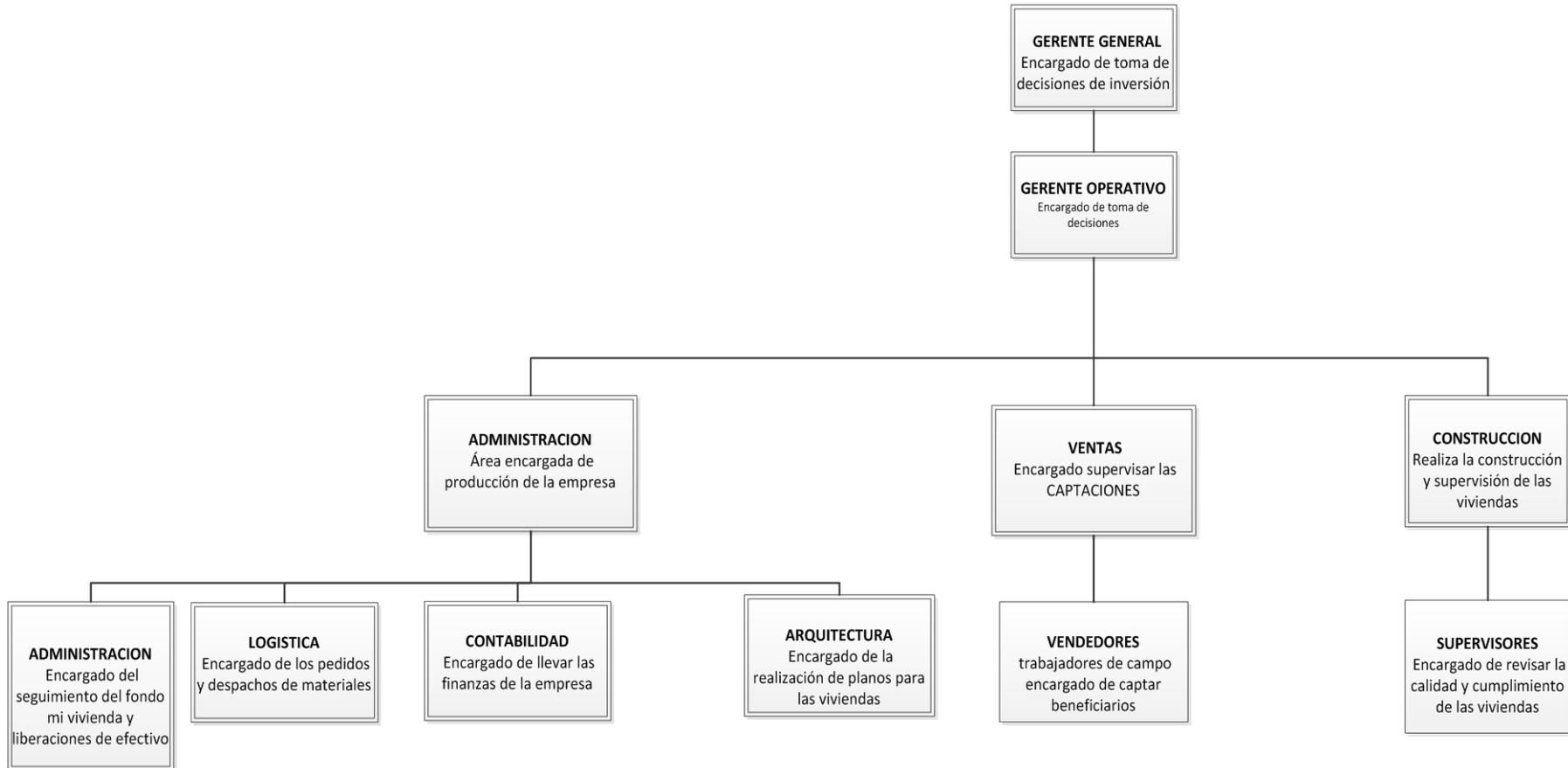
### 1.1.8.2. Organigrama nominal

Img.4: Organigrama nominal.



### 1.1.8.3. Organigrama funcional

Img.5: Organigrama funcional



### 1.1.9. Hardware y software

Tbl.1: Hardware

<b>HARDWARE</b>	
<b>ADMINISTRACION</b>	<b>CANTIDAD</b>
<b>LAPTOP</b>	<b>8</b>
Sistema operativo:	Microsoft Windows 8 (x64) Build 9200
Procesador:	AMD E1-1500 APU (3700.0 MHz)
Placa base:	(AMD A55/A60M (Hudson-D2/M2))
Memoria:	2 GBytes (2 GBytes)
<b>COMPUTADORA</b>	<b>1</b>
Sistema operativo:	Microsoft Windows XP (X86)
<b>IMPRESORA</b>	<b>1</b>
	RICOH AFICIO 500 MP
<b>PROYECTOR MULTIMEDIA</b>	<b>1</b>
<b>CONSTRUCCION</b>	<b>CANTIDAD</b>
<b>LAPTOP</b>	<b>3</b>
Marca del equipo:	Hewlett-Packard HP 425
Sistema operativo:	Microsoft Windows 7 Home Basic Build 7601
Service Pack:	Service Pack 1
Procesador:	AMD Athlon II P320 (2100.0 MHz)
<b>IMPRESORA</b>	<b>1</b>
	RICOH AFICIO 500 MP
<b>ARQUITECTURA</b>	<b>CANTIDAD</b>
<b>LAPTOP</b>	<b>3</b>
Marca del equipo:	GIGABYTE
Sistema operativo:	Microsoft Windows 8 Professional (x64) Build 9200
Procesador:	Intel Core i5-3330 (3000.0 MHz)
Placa base:	GIGABYTE B75M-D3H (Intel B75 (Panther Point-M Enhanced))
<b>COMPUTADORA</b>	<b>2</b>
Sistema operativo:	Microsoft Windows 8 Professional (x64) Build 9200
Procesador:	Intel Core i5-3330 (3000.0 MHz)
<b>IMPRESORA</b>	<b>1</b>
	RICOH AFICIO 500 MP

Tbl.2: Software

SOFTWARE		
ÍTEM	DESCRIPCIÓN	LICENCIA
Microsoft Office 2010	paquete de programas ofimáticos (Excel, Word, PowerPoint, Access) para oficina desarrollado por Microsoft Corp.	NO
AutoCad 2010	Es un programa de dibujo técnico desarrollado por Autodesk para el uso de ingenieros, técnicos y otros profesionales de carreras de diseño	NO
Eset Nod32 6	Antivirus	NO

## 1.2. Situación problemática

En la gran mayoría de instituciones el trámite documentario se realiza de forma manual. A continuación se describen algunas consecuencias de esta práctica:

- Se deben realizar gastos adicionales para cubrir los costos del abundante material de escritorio empleado.
- Existe una sola copia del expediente de un trámite, que es transportada físicamente, en cada etapa del flujo. En la mayoría de casos el flujo es inexistente, y el camino se decide en cada proceso.
- Si el personal asignado a un trámite desea revisar un expediente debe esperar a que éste esté disponible. Incluso algunas

instituciones, destinan personal exclusivamente al traslado de documentos. También existe el problema potencial de la alteración deliberada de documentos.

- No es fácil detectar en qué están ocupados los empleados de cada área ni las razones de los retrasos. Todo ello producto de la falta de control de esta forma de administrar los documentos.

Todos los problemas mencionados son la razón principal de la existencia de este sistema.

### **1.3. Problema**

¿Se logrará mediante la sistematización de los procesos una gestión e integridad de información, entre las áreas de administración y construcción de la empresa constructora "AIF CONSTRATISTAS GENERALES S.A.C"?

### **1.4. Hipótesis**

Mediante la sistematización de procesos se lograra la gestión de la información que influye significativamente para la correcta integridad en los trámites documentarios y control entre las áreas de administración y construcción de la empresa constructora "AIF CONTRATISTAS GENERALES S.A.C".

### **1.5. Objetivos**

#### **1.5.1. Objetivo general**

La gestión de la información para lograr salvaguardar la integridad y reutilización de los datos dentro de procesos de trámites documentarios y auditorios de la empresa constructora "AIF CONSTRATISTAS GENERALES S.A.C".

### **1.5.2. Objetivos específicos**

- ✓ Control sobre cantidades de casas construidas y en construcción.
- ✓ Minimizar tiempos en los procesos.
- ✓ Centralización de la data.
- ✓ Mejorar tiempos en el proceso documentario
- ✓ Control de crecimiento para la toma de decisiones
- ✓ Mejora en la aceleración de documentos para trámites
- ✓ Simplificar los procesos para los usuarios
- ✓ Lograr tener un sistema integrado.
- ✓ Brindar un mejor servicio a los clientes.

### **1.6. Justificación e importancia**

El proyecto se justifica en la presente escasa comunicación interna en cada área y la falta de relación entre las mismas con respecto a la información, con una descentralización de la data que significa un gran desbalance en la auditoría anual.

La sistematización de procesos se hace para la gestión de la información documentaria entre las áreas de administración y construcción de la empresa constructora "AIF CONSTRATISTAS GENERALES S.A.C" tomará la importancia de conseguir que todos los documentos estén salvaguardados, logrando la gestión de los procesos y poder tener resultados reales con la utilización del control en cada cierre de año.

Mediante la creación e implementación de un sistema diseñado a medida con aplicación web y de escritorio.

## 1.7. Metodología

### 1.7.1. Modo de investigación

- ✓ Multidisciplinario

### 1.7.2. Tipo de investigación

- ✓ Tecnológica – Formal

Incluye los campos de la programación de computadoras, el análisis de sistemas y los métodos que en su conjunto forman la investigación operativa.

### 1.7.3. Contrastación de hipótesis

#### 1.7.3.1. Modelo de contrastación

##### 1.7.3.1.1. Consecuencias lógicas

Si se desarrolla el Sistema de Sistematización de Procesos para la gestión de la Información Documentaria se logrará personas satisfechas de su trámite.

##### 1.7.3.1.2. Operacionalización de variables

Tbl.3: Definición de la variable independiente

VARIABLE INDEPENDIENTE	DEFINICIÓN	INDICADORES
Sistema de Informático	Sistema que se encarga de sistematizar los procesos y de la gestión de la información de expedientes y organización de la	<ul style="list-style-type: none"> <li>✓ Número de expedientes tramitados por año.</li> <li>✓ Conexión, usabilidad y mantenimiento</li> </ul>

	institución en la que se implementa, haciendo uso de tecnologías.	
--	---	--

Tbl.4: Definición de variable dependiente

VARIABLE DEPENDIENTE	DEFINICIÓN	INDICADORES
Sistematizar los procesos y la gestión de la información de expedientes de la Constructora	Sistematización de un proceso previamente definido, con la finalidad de normalizarlo y organizarlo, obteniendo como resultado una aplicación informática que responde a todos los parámetros y premisas que se formularon y diseñaron	<p>✓ Porcentaje de reducción del tiempo de espera por remitente.</p> <p>✓ Porcentaje de aumento en cantidad de expedientes procesados por día.</p>

#### 1.7.4. POBLACIÓN Y MUESTRA ESTADÍSTICA DE ESTUDIO

De acuerdo a la información proporcionada por la empresa AIF Contratistas Generales, el número de posibles usuarios del sistema es 30. A partir de esta población se pasa a determinar la muestra, la cual se hizo mediante el cálculo de muestra para una población finita haciendo uso de la siguiente fórmula:

$$n = \frac{z^2 * N * p * q}{(n - 1) * e^2 + z^2 * p * q}$$

**Dónde:**

Tamaño de la muestra: n

Tamaño de la población: N

Probabilidad de éxito: q

Margen de error: E

Nivel de confianza requerido: Z

Para el estudio se tomaran en cuenta las siguientes restricciones:

p= 0.5

q= 0.5

E= 9% = 0.09

Z= 1.96

**Reemplazando: n=24**

Cuando la población es pequeña, la muestra obtenida mediante esta fórmula es demasiado grande, en estos casos se debe aplicar la siguiente formula correctora:

$$nh = Nh(n)/N$$

**Dónde:**

Tamaño de la población : N=30

Tamaño de la muestra total : n= 24

Número de jefes de área : Nh= 7

**Reemplazando: nh =5**

## **1.7.5. Técnica e instrumento de recopilación de datos**

### **1.7.5.1. Entrevista:**

Las entrevistas se utilizan para recopilar información en forma verbal, a través de preguntas que propone el grupo de investigación. Quienes responden pueden ser los jefes o empleados de cada área, en otras palabras, la entrevista es un intercambio de información que se efectúa cara a cara. Es un canal de comunicación entre el entrevistador y los usuarios. La entrevista es la técnica más significativa y productiva de que dispone el grupo investigador para recabar datos. Por otra parte, la entrevista ofrece al investigador una excelente oportunidad para establecer una conversación con el personal de trabajo, lo cual es fundamental en el transcurso del tema de investigación.

## **2. CAPITULO II**

### **MARCO TEÓRICO**

## **2.1. Revisión bibliográfica**

### **2.1.1. Sistema de gestión documental para la organización de los proyectos y expedientes desarrollados en el área de dirección de estudios de la dirección ejecutiva del proyecto especial olmos tinajones – Depolti Chiclayo (UNPRG – Lambayeque).**

#### **Autores:**

- ✓ Bach. Luisa Orfelinda Arenas Llontop.
- ✓ Bach. Veronica Elizabeth Cornetero Muro.

**Año:** 2005

**Tipo:** Tesis

#### **Resumen**

El presente trabajo de investigación se desarrolló ante las diferentes necesidades de gestión de información encontradas en la Oficina de Dirección de Estudios, Desarrollo Agrícola y Medio Ambiente de la Dirección Ejecutiva del Proyecto Especial Olmos Tinajones – DEPOLTI; trayendo como consecuencia la propuesta de implementar un sistema de gestión documental en dicha oficina, siendo uno de los primeros proyectos que se realizan en esta institución orientados a la gestión y control de documentos que allí se elaboran.

### **2.1.2. Análisis y diseño de un sistema de trámite de documentos de pago a proveedores vía intranet – PUCP de Lima.**

**Autor:** Dorila Sarita Carrera Jiménez

**Año:** 2009

**Tipo:** Tesis

## Resumen

La presente tesis propone el análisis y diseño de un sistema de trámite de documentos de pago a proveedores vía Intranet, que puede ser implementado en cualquier institución organizada en unidades.

- La organización de este documento, guía al lector en el conocimiento gradual del problema, el análisis y diseño de la alternativa de solución. Así, en el primer capítulo, se presenta la definición y el marco conceptual del problema, y se describe y sustenta la alternativa de solución. En el segundo capítulo, se detalla la metodología a utilizar, los requerimientos identificados y el análisis de los mismos.
- En el tercer capítulo, se diseña la alternativa de solución. Finalmente, en el cuarto capítulo, se incluyen las observaciones, conclusiones y recomendaciones.
- A manera de aplicación práctica se presentan el análisis y diseño para la Pontificia Universidad Católica del Perú, institución que recibe un promedio diario de cuatrocientos cincuenta documentos de sus proveedores, los cuales son consecuencia de la adquisición de bienes y servicios que realizan las diferentes unidades que la conforman.

### **2.1.3. Desarrollo de un sistema para el registro y control de la documentación relacionada con el presupuesto de las obras civiles en ejecución por la empresa atina IPC - Universidad Alejandro de Humboldt de Caracas.**

**Autor:** Iván Schartun

**Año:** 2009

**Tipo:** Tesis

#### **Resumen**

La presente de investigación tuvo como propósito Desarrollar un sistema para el registro y control de la documentación relacionada con el presupuesto de las obras civiles en ejecución por la empresa ATINA IPC C.A. La metodología quedó enmarcada en un trabajo de campo, no experimental y bajo la modalidad de proyecto factible. La población objeto de estudio fue finita, es decir, seis (06) usuarios finales y por tanto, lo que no ameritó la aplicación de criterio matemático para la establecer la muestra, a los cuales se le aplicó la técnica de la encuesta y el instrumento del cuestionario con preguntas abiertas y cerradas. Dicho instrumento fue validado por juicio de experto. El análisis de la información recogida fue cuanti-cualitativo y permitió plantear las siguientes conclusiones: El proceso de control y registro de la documentación de la empresa ATINA IPC se lleva de manera manual, y por lo tanto, presenta la necesidad de un sistema automatizado para el control y registro de la información que se maneja en dicha empresa y es factible desde un punto de vista técnico el desarrollo de un sistema de registro y control en la empresa ATINA IPC, con la finalidad de automatizar los procesos a través del sistema, entre otras.

#### **2.1.4. Análisis, diseño e implementación de un sistema de solicitudes de pago de remuneraciones eventuales vía una intranet – PUCP de Lima.**

**Autor:** María Isabel Carranza Liza.

**Año:** 2009.

**Tipo:** Tesis.

##### **Resumen**

La presente tesis propone el análisis, diseño e implementación de un sistema de solicitudes de pago de remuneración adicional eventual vía una Intranet para la Pontificia Universidad Católica del Perú. Esta aplicación estará disponible para el personal de las unidades, de la oficina de Presupuesto y de la sección Planilla que tengan los permisos de acceso necesarios.

El presente documento de tesis se encuentra dividido en capítulos, y en cada uno de ellos se explican las actividades desarrolladas durante las fases de análisis, diseño e implementación del sistema planteado.

En el primer capítulo de este documento de tesis se plantean la definición y marco conceptual del problema, y se describe y sustenta la alternativa de solución presentada. En el segundo capítulo se explica y justifica la metodología utilizada. Asimismo, se identifican y explican los requerimientos del sistema. En el tercer capítulo se presenta el diseño de la solución propuesta. En el cuarto capítulo se describen y justifican las herramientas utilizadas en la fase

de implementación de la solución, y se presenta el plan de pruebas realizado al sistema. Y Finalmente, en el quinto capítulo se mencionan observaciones, conclusiones y recomendaciones para trabajos futuros relacionados al tema planteado en este tema de tesis.

#### **2.1.5. Sistema web de consultas para la gestión de trámite documentario de la municipalidad provincial de Sullana- Piura”**

**Autor:** Carlos Andrés Medina Sandoval

**Año:** 2007.

**Tipo:** Tesis.

#### **RESUMEN**

Este trabajo de tesis se realizó con el objetivo de mejorar la gestión de trámite documentario, con especial énfasis en las consultas realizadas antes y durante la tramitación de documentos de importancia presentados por los ciudadanos y recepcionado por la Unidad de Trámite Documentario y Archivo. Se investigó en la Municipalidad Provincial de Sullana, Departamento Piura. La zona pertenece al Distrito de Sullana. (Gobierno de Sullana, 2003). La afluencia de ciudadanos que visitan directamente la Unidad de Trámite Documentario es en promedio mensual de 3,812 personas, ya sea para consultas recepción o entrega de documentos. Las muestras usadas dentro de la investigación permitieron extraer información de la problemática antes y después de la solución a implantar.

### **2.1.6. Diseño de sistema de trámites documentarios oficiales nacional**

**Autor:** Giancarlo André Gonzales Vargas

**Año:** 2010.

**Tipo:** Tesis.

#### **RESUMEN**

En la actualidad, miles de personas por día necesitan acudir a una entidad pública del Estado para obtener un documento oficial de la misma. Los motivos son varios, y cuando hay que realizarlo, se hace indefectiblemente.

Cuando se necesita un documento (o varios). Previamente, hay que realizar el pago de una tasa al Banco de la Nación u otra entidad bancaria con la que el Estado interactúe y juntar algunos requisitos adicionales como fotografías tamaño pasaporte y fotocopias del DNI. El proceso, como se puede apreciar, es largo y, a veces, incómodo, tanto antes, como durante la realización de ésta.

Viviendo en el siglo XXI, donde la automatización de procesos, la digitalización de la información y la masificación de las telecomunicaciones ya no son una innovación, sino una forma eficiente de realizar tareas cotidianas en nuestra sociedad.

## 2.2. Base teórica

### 2.2.1. Las metodologías ágiles

#### 2.2.1.1. Los principales valores de las metodologías ágiles

**2.2.1.1.1. Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas:** La gente es el principal factor de éxito de un proyecto de software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.

**2.2.1.1.2. Desarrollar software que funciona más que conseguir una buena documentación:** La regla a seguir es "no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante". Estos documentos deben ser cortos y centrarse en lo fundamental.

**2.2.1.1.3. La colaboración con el cliente más que la negociación de un contrato:**

Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

#### **2.2.1.1.4. Responder a los cambios más que seguir**

**estrictamente un plan:** La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

#### **2.2.1.2. Principios**

- ✓ La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- ✓ Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- ✓ Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- ✓ La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- ✓ Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- ✓ El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.

- ✓ El software que funciona es la medida principal de progreso.
- ✓ Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- ✓ La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- ✓ La simplicidad es esencial.
- ✓ Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- ✓ En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

### **2.2.1.3. Metodologías ágiles versus metodologías tradicionales**

Tener metodologías diferentes para aplicar de acuerdo con el proyecto que se desarrolle resulta una idea interesante. Estas metodologías pueden involucrar prácticas tanto de metodologías ágiles como de metodologías tradicionales. De esta manera podríamos tener una metodología para cada proyecto, la problemática sería definir cada una de las prácticas, y en el momento preciso definir parámetros para saber cuál usar.

Es importante tener en cuenta que el uso de un método ágil no es para todos. Sin embargo, una de las principales ventajas de los métodos ágiles es su peso inicialmente ligero y por eso las personas que no estén acostumbradas a seguir procesos encuentran estas metodologías bastante agradables.

Tbl.5: Diferencias entre Metodologías Ágiles y no Ágiles.

<b>Metodologías Ágiles</b>	<b>Metodologías Tradicionales</b>
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
Impuestas internamente (por el equipo).	Impuestas externamente.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

#### **2.2.1.4. ¿Por qué usar metodologías ágiles?**

Las metodologías tradicionales presentan los siguientes problemas a la hora de abordar proyectos:

- ✓ Existen unas costosas fases previas de especificación de requisitos, análisis y diseño. La corrección durante el desarrollo de errores introducidos en estas fases será costosa, es decir, se pierde flexibilidad ante los cambios.
- ✓ El proceso de desarrollo está encorsetado por documentos firmados.
- ✓ El desarrollo es más lento. Es difícil para los desarrolladores entender un sistema complejo en su globalidad.

Las metodologías ágiles de desarrollo están especialmente indicadas en proyectos con requisitos poco definidos o cambiantes. Estas metodologías se aplican bien en equipos pequeños que resuelven problemas concretos, lo que no está reñido con su aplicación en el desarrollo de grandes sistemas, ya que una correcta popularización de los mismos es fundamental para su exitosa implantación. Dividir el trabajo en módulos abordables minimiza los fallos y el coste.

#### **2.2.1.5. Metodologías ágiles de desarrollo de software**

##### **2.2.1.5.1. Xp-extreme programming**

Básicamente se podría decir que la programación extrema es una "metodología ligera o ágil" para el desarrollo de software eficiente y altamente efectivo.

Como metodología, la programación extrema, presenta muchos puntos comunes con el desarrollo incremental, comenzando por el hecho de que el software desarrollado con XP se realiza de forma incremental.

#### **2.2.1.5.1.1. Las historias de usuario**

Es la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Contenidos de la ficha:

- ✓ Fecha.
  
- ✓ Tipo de actividad (nueva, corrección, mejora).
  
- ✓ Prueba funcional.
  
- ✓ Número de historia.
  
- ✓ Prioridad técnica y del cliente.
  
- ✓ Referencia a otra historia previa
  
- ✓ Riesgo.

- ✓ Estimación técnica.
- ✓ Descripción.
- ✓ Notas.
- ✓ Lista de seguimiento con la fecha, estado, cosas por terminar y comentarios.

A efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración). Las historias de usuario son descompuestas en tareas de programación (task card) y asignadas a los programadores para ser implementadas durante una iteración.

#### **2.2.1.5.1.2. Proceso xp**

El ciclo de desarrollo consiste en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.

4. El programador construye ese valor de negocio.

5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Reléase), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

#### **2.2.1.5.1.3. Prácticas xp**

- ✓ **Planificación:** se utilizan las "user-stories" ("historias del usuario"), para realizar el análisis, estas "historias", se dividirán en tareas (unidades pequeñas, de 1 a 5 días de trabajo en pareja). Además, se priorizarán las tareas, y cada una de ellas tendrá un desarrollo incremental.

- ✓ **Versiones pequeñas:** La primera versión contendrá el conjunto mínimo de requisitos más útiles/necesarios para el sistema global.
  
- ✓ **Sistema metafórico:** Cada proyecto debe tener una metáfora asociada que nos ofrezca unos criterios para nombrar lo que vayamos haciendo de forma fácil.
  
- ✓ **Diseño simple:** Como los requerimientos cambian, o pueden hacerlo, diariamente, hay que utilizar los diseños más simples posibles para cumplir los requerimientos que tenemos en la actualidad.
  
- ✓ **Testeo continuo:** Antes de que se implemente cualquier característica de un sistema, se debe escribir un test para ella.
  
- ✓ **Refactoring:** Cuando tenemos que introducir una nueva característica del sistema, si esta tiene mucho en común con otra previa, lo mejor es eliminar el código duplicado, sin miedo a que falle, debido a que el test probará el correcto funcionamiento.
  
- ✓ **Pair programming ("programación en parejas"):** Se trabaja en parejas, cada una utilizando un único ordenador. Así, el código se revisa mientras se desarrolla.

- ✓ **Propiedad colectiva del código:** Cualquiera puede modificar cualquier módulo en cualquier momento, nadie tiene la propiedad de ningún módulo.
- ✓ **Integración continua:** Todos los cambios se introducen en el sistema, al menos, una vez al día.
- ✓ **Semanas de 40 horas de trabajo:** Los programadores se deben ir a casa a su hora.
- ✓ **Cliente en su sitio:** Siempre hay un usuario del sistema que es accesible por los miembros del equipo de trabajo.
- ✓ **Estándares de codificación:** Todos deben usar los mismos criterios a la hora de programar.

#### 2.2.1.5.2. Scrum

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a

cabo en un periodo de tiempo breve (normalmente de 30 días).

Cada uno de estos periodos de desarrollo es una iteración que finaliza con la producción de un incremento operativo del producto.

Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente.

#### **2.2.1.5.2.1. Control de la evolución del proyecto**

Scrum controla de forma empírica y adaptable la evolución del proyecto, empleando las siguientes prácticas de la gestión ágil:

##### **✓ Revisión de las Iteraciones**

Al finalizar cada iteración (normalmente 30 días) se lleva a cabo una revisión con todas las personas implicadas en el proyecto. Este es el periodo máximo que se tarda en reconducir una desviación en el proyecto o en las circunstancias del producto.

##### **✓ Desarrollo incremental**

Durante el proyecto, las personas implicadas no trabajan con diseños o abstracciones. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.

#### ✓ **Desarrollo evolutivo**

Los modelos de gestión ágil se emplean para trabajar en entornos de incertidumbre e inestabilidad de requisitos.

Intentar predecir en las fases iniciales cómo será el producto final, y sobre dicha predicción desarrollar el diseño y la arquitectura del producto no es realista, porque las circunstancias obligarán a remodelarlo muchas veces.

Para qué predecir los estados finales de la arquitectura o del diseño si van a estar cambiando. En Scrum se toma a la inestabilidad como una premisa, y se adoptan técnicas de trabajo para permitir esa evolución sin degradar la calidad de la arquitectura que se irá generando durante el desarrollo.

El desarrollo Scrum va generando el diseño y la arquitectura final de forma evolutiva durante todo el proyecto. No los considera como productos que deban realizarse en la primera "fase" del proyecto.

✓ **Auto-organización**

En Scrum los equipos son auto-organizados (no auto-dirigidos), con margen de decisión suficiente para tomar las decisiones que consideren oportunas.

✓ **Colaboración**

Las prácticas y el entorno de trabajo ágiles facilitan la colaboración del equipo. Ésta es necesaria, porque para que funcione la auto-organización como un control eficaz cada miembro del equipo debe colaborar de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

**2.2.1.5.2.2. Visión general del proceso**

Scrum denomina "sprint" a cada iteración de desarrollo y recomienda realizarlas con duraciones de 30 días.

El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental.

Los elementos que conforman el desarrollo Scrum son:

### ✓ **Las reuniones**

- Planificación de sprint: Jornada de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.
- Reunión diaria: Breve revisión del equipo del trabajo realizado hasta la fecha y la previsión para el día siguiente.
- Revisión de sprint: Análisis y revisión del incremento generado.

### ✓ **Los elementos**

- Pila del producto: lista de requisitos de usuario que se origina con la visión inicial del producto y va creciendo y evolucionando durante el desarrollo.
- Pila del sprint: Lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- Incremento: Resultado de cada Sprint.

### ✓ **Los roles**

Scrum clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto en: propietario del producto,

equipo, gestor de Scrum (también Scrum Manager o Scrum Master) y “otros interesados”. Los tres primeros grupos (propietario, equipo y gestor) son los responsables del proyecto, los que según la comparación siguiente (y sin connotaciones peyorativas) serían los “cerdos”; mientras que el resto de interesados serían las gallinas.

### 2.2.1.5.2.3. Cerdos y gallinas.

Esta metáfora ilustra de forma muy gráfica la diferencia de implicación en el proyecto entre ambos grupos:

*Una gallina y un cerdo paseaban por la carretera.*

*La gallina dijo al cerdo: “Quieres abrir un restaurante conmigo”.*

*El cerdo consideró la propuesta y respondió: “Sí, me gustaría. ¿Y cómo lo llamaríamos?”.*

*La gallina respondió: “Huevos con beicon”.*

*El cerdo se detuvo, hizo una pausa y contestó:*

*“Pensándolo mejor, creo que no voy a abrir un restaurante contigo. Yo estaría realmente comprometido, mientras que tu estarías sólo implicada”.*

Tbl.6: Los roles de Scrum

COMPROMETIDOS (cerdos)	IMPLICADOS (gallinas)
✓ Propietario del producto	✓ Otros interesados (Dirección General,
✓ Equipo	Dirección Comercial,

✓ Scrum Master (Marketing, Usuarios, etc.)

- Propietario del producto: El responsable de obtener el mayor valor de producto para los clientes, usuarios y resto de implicados.
- Equipo de desarrollo: grupo o grupos de trabajo que desarrollan el producto.
- Scrum Manager: gestor de los equipos que es responsable del funcionamiento de la metodología Scrum y de la productividad del equipo de desarrollo.

✓ **Valores**

Scrum es una "carrocería" para dar forma a los principios ágiles. Es una ayuda para organizar a las personas y el flujo de trabajo.

La carrocería sin motor, sin los valores que dan sentido al desarrollo ágil, no funciona.

- Delegación de atribuciones (empowerment) al equipo para que pueda auto-organizarse y tomar las decisiones sobre el desarrollo.
- Respeto entre las personas. Los miembros del equipo deben confiar entre ellos y respetar sus conocimientos y capacidades.

- Responsabilidad y auto-disciplina (no disciplina impuesta).
- Trabajo centrado en el desarrollo de lo comprometido.
- Información, transparencia y visibilidad del desarrollo del proyecto.

### **2.2.1.5.3. Rad**

La metodología de desarrollo conocida como diseño rápido de aplicaciones RAD (por sus siglas en inglés) ha tomado gran auge debido a la necesidad que tienen las instituciones de crear aplicaciones funcionales en un plazo de tiempo corto. Esta modalidad de desarrollo consiste de diferentes etapas que suceden de forma paralela y exigen la colaboración de los usuarios en todos los niveles. Por el contrario, en la metodología de diseño tradicional, las etapas suceden de forma lineal y el usuario es excluido del proceso, lo que hace que esta modalidad sea más lenta y poco eficiente.

#### **2.2.1.5.3.1. La metodología de diseño rápido de aplicaciones**

Esta metodología propone un proceso de desarrollo de "**software**" que permite que se creen sistemas de computadoras utilizables en un periodo de tiempo entre 60 a 90 días. RAD es un ciclo de desarrollo diseñado para crear aplicaciones de computadoras de alta calidad

de las que acontecen en corporaciones grandes.

El desarrollo de aplicaciones enfrenta una transformación fundamental. Hace cinco años un proyecto para desarrollar una aplicación tomaba un periodo de entre 18 a 24 meses; actualmente, con la práctica del modelo RAD toma entre 1 a 3 meses.

#### **2.2.1.5.3.2. Las cuatro etapas del ciclo rad**

##### **✓ Etapa de planificación de los requisitos**

Esta etapa requiere que usuarios con un vasto conocimiento de los procesos de la compañía determinen cuáles serán las funciones del sistema. Debe darse una discusión estructurada sobre los problemas de la compañía que necesitan solución

##### **✓ Etapa de diseño**

Esta consiste de un análisis detallado de las actividades de la compañía en relación al sistema propuesto. En ellos descomponen funciones y definen entidades asociadas con el sistema. Una vez se completa el análisis se crean los diagramas que definen las alteraciones entre los procesos y la data. Al finalizar el análisis se traza el diseño del

sistema. Se desarrollan los procedimientos y los esquemas de pantallas.

#### ✓ **Construcción**

En la etapa de construcción el equipo de desarrolladores trabajando de cerca con los usuarios finaliza el diseño y la construcción del sistema. La construcción de la aplicación consiste de una serie de pasos donde los usuarios tienen la oportunidad de afirmar los requisitos y repasar los resultados. Las pruebas al sistema se llevan a cabo durante esta etapa. También se crea la documentación y las instrucciones necesarias para manejar la nueva aplicación, rutinas y procedimientos para operar el sistema.

#### ✓ **Implementación**

Esta etapa envuelve la implementación del nuevo producto y el manejo del cambio del viejo al nuevo sistema. Se hacen pruebas comprensivas y se adiestran los usuarios. Los cambios organizacionales y la operación del nuevo sistema se hacen en paralelo con el viejo sistema hasta que el nuevo se establezca completamente.

### 2.2.2. Microsoft .net

Es una plataforma de desarrollo y ejecución de aplicaciones. Esto quiere decir que no sólo nos brinda todas las herramientas y servicios que se necesitan para desarrollar modernas aplicaciones empresariales y de misión crítica, sino que también nos provee de mecanismos robustos, seguros y eficientes para asegurar que la ejecución de las mismas sea óptima. Los componentes principales de la plataforma .NET son:

- ✓ Un entorno de ejecución de aplicaciones, también llamado "Runtime", que es un componente de software cuya función es la de ejecutar las aplicaciones .NET e interactuar con el sistema operativo ofreciendo sus servicios y recursos.
- ✓ Un conjunto de bibliotecas de funcionalidades y controles reutilizables, con una enorme cantidad de componentes ya programados listos para ser consumidos por otras aplicaciones.
- ✓ Un conjunto de lenguajes de programación de alto nivel, junto con sus compiladores y linkers, que permitirán el desarrollo de aplicaciones sobre la plataforma .NET.
- ✓ Un conjunto de utilitarios y herramientas de desarrollo para simplificar las tareas más comunes del proceso de desarrollo de aplicaciones
- ✓ Documentación y guías de arquitectura, que describen las mejores prácticas de diseño, organización, desarrollo, prueba e instalación de aplicaciones .NET

En la ilustración podemos ver que la plataforma Microsoft.NET es denominada "de Ejecución Intermedia" justamente porque se ubica entre el Sistema Operativo y las aplicaciones finales con las que interactúan los usuarios, actuando como intermediario entre ambos.

Img.6: Plataforma Microsoft .net



#### 2.2.2.1. Ventajas de .net

- ✓ Unifica los modelos de programación.
- ✓ Simplifica aún más el desarrollo.
- ✓ Provee un Entorno de Ejecución robusto y seguro.
- ✓ Es independiente del lenguaje de programación.
- ✓ Interoperabilidad con código existente.
- ✓ Simplifica la instalación y administración de las aplicaciones.
- ✓ Es Extensible.

## 2.2.2.2. Componentes fundamentales

### 2.2.2.2.1. Net framework

El .NET Framework (traducido como "Marco de Trabajo") es el componente fundamental de la plataforma Microsoft .NET, necesario tanto para poder desarrollar aplicaciones como para poder ejecutarlas luego en entornos de prueba o producción.

### 2.2.2.2.2. Common language runtime(CLR)

Veamos ahora algunos de los principales servicios que les brinda a las aplicaciones:

✓ **Compilación Just In Time (o Justo A Tiempo):**

El CLR se encarga de compilar las aplicaciones .NET a código de máquina nativo para el sistema operativo y la plataforma de hardware en la que se está ejecutando.

✓ **Gestión Automática de Memoria:**

El CLR abstrae a los desarrolladores de tener que pedir y liberar memoria explícitamente. Para esto, uno de sus componentes llamado Garbage Collector (Recolector de Basura) se encarga de liberar periódicamente la memoria que ya no está siendo usada por ninguna aplicación.

✓ **Gestión de Errores Consistente:**

Como las aplicaciones .NET no se ejecutan directamente contra el Sistema Operativo, cualquier error no manejado que ocurra en tiempo de ejecución será atrapado por el CLR en última instancia, no afectando a ninguna otra aplicación que se esté ejecutando ni teniendo efecto alguno sobre su estabilidad.

✓ **Ejecución Basada en Componentes:**

Todas las aplicaciones .NET son empaquetadas en componentes reutilizables denominados genéricamente Assemblies, que el CLR se encarga de cargar en memoria y ejecutar.

**2.2.2.2.3. Class library**

- ✓ Conjunto de Tipos básicos (clases, interfaces, etc.) que vienen incluidos en el .NET Framework.
- ✓ Los tipos están organizados en jerarquías lógicas de nombres, denominados NAMESPACES.
- ✓ Los tipos son independientes del lenguaje de desarrollo.
- ✓ Es extensible y totalmente orientada a objetos.

Img.7: Namespace SYSTEM

El namespace raíz es **SYSTEM**



Los principales namespaces de la biblioteca de clases .NET son:

- ✓ System: raíz de todos los otros namespaces, y dentro del cual podemos encontrar la mayoría de los namespaces correspondientes a la Base Class Library.
- ✓ System.Data y System.Xml: en conjunto, estos dos namespaces constituyen la tecnología conocida como ADO.NET.
- ✓ System.Web: dentro de este namespace se encuentran todos los tipos necesarios para programar aplicaciones y servicios web ASP.NET.
- ✓ System.Windows.Forms: dentro de este namespace se encuentran todos los tipos necesarios para programar aplicaciones de escritorio basadas en formularios y ventanas Windows.

#### **2.2.2.2.4. Common language specification (CLS)**

Uno de los objetivos de diseño de la plataforma .NET fue el ser independiente del lenguaje de programación elegido para el desarrollo de aplicaciones. Para lograr esto es que se creó la Especificación de Lenguaje Común (o CLS, por sus siglas en inglés), que define y estandariza un subconjunto de todas las características soportadas por el CLR y que son necesarias en la mayoría de las aplicaciones.

Junto con el .NET Framework, Microsoft provee implementaciones de cuatro lenguajes compatibles con CLS, junto con sus compiladores:

- ✓ Microsoft Visual Basic .NET
- ✓ Microsoft Visual C# .NET
- ✓ Microsoft Visual J#.NET
- ✓ Microsoft Visual C++.NET

#### **2.2.2.3. Bibliotecas principales**

##### **2.2.2.3.1. Base class library**

La Base Class Library constituye los fundamentos de la .NET Framework Class Library, ya que provee la mayor parte de las funcionalidades elementales que pueden necesitarse para construir una aplicación o servicio.

En la ilustración4 se pueden apreciar los namespaces más importantes que componen la BCL.

Img.8: Principales Namespace de BCL

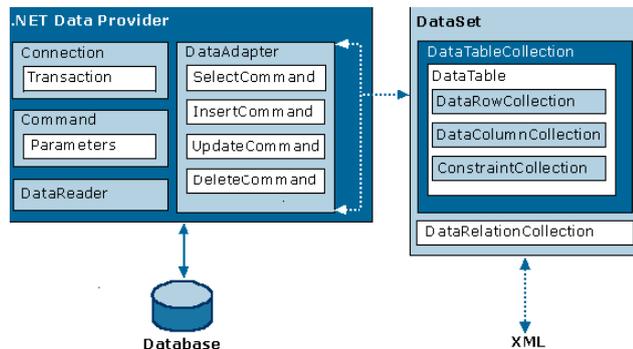


## 2.2.2.3.2. Ado.net

### 2.2.2.3.2.1. Ado.net – arquitectura

La arquitectura de ADO.NET está basada en el concepto de proveedores de acceso a datos, siendo un proveedor un conjunto de clases que permiten conectarse a una base de datos, ejecutar un comando sobre ella y tener acceso a los resultados de su ejecución, tanto de forma conectada como desconectada.

Img.9: Arquitectura ADO .NET



ADO.NET proporciona nuevas características para Entity Framework, incluidos objetos que ignoran la persistencia, funciones en consultas LINQ y generación de código personalizado de la capa de objeto.

#### ✓ **Datos Dinámicos**

En ASP.NET 4, se ha mejorado los datos dinámicos para ofrecerle aún más eficacia a la hora de compilar rápidamente sitios web controlados por datos. Entre estas estructuras se incluyen las siguientes:

- Validación automática basada en restricciones que se definen en el modelo de datos.
- Posibilidad de cambiar fácilmente el código que se genera para campos en los controles DetailsView y GridView usando plantillas de campo que forman parte de un proyecto de Datos dinámicos.

#### ✓ **Servicios de datos de WCF**

Servicio de datos de ADO.NET se denomina ahora Servicios de datos de WCF y tiene las siguientes características nuevas:

- Enlace de datos.
- Contar las entidades de un conjunto de entidades.

- Paginación controlada por servidor.
- Proyecciones de consultas.
- Proveedores de servicios de datos personalizados.
- Transmitir por secuencias recursos binarios.

#### ✓ **Windows Communication Foundation**

Proporciona las mejoras siguientes:

- Integración de System.Web.Routing: le ofrece más control sobre la dirección URL del servicio, al permitir el uso de direcciones URL sin extensión.
- Compatibilidad con varios enlaces de sitios de IIS: le permite tener varias direcciones base con el mismo protocolo en el mismo sitio web.
- Servicio de enrutamiento: le permite enrutar mensajes según el contenido.
- Extremos estándar: extremos predefinidos que le permiten especificar únicamente ciertas propiedades.

### 2.2.2.3.3. Asp.net

ASP.NET es un subconjunto de la .NET Framework Class Library que contiene las funcionalidades necesarias para desarrollar aplicaciones y servicios Web, y sus clases se encuentran dentro del namespace System.Web. ASP.NET no es sólo la nueva versión de su predecesor, ASP, sino que provee un nuevo modelo unificado de programación orientada a objetos que permite hacer uso de todos los servicios y facilidades del .NET Framework programando en cualquier lenguaje compatible con la plataforma. Por otra parte, nuevos servicios a nivel de infraestructura (seguridad, performance, estabilidad, configuración, instalación, mantenimiento) hacen que ASP.NET sea ideal para construir aplicaciones web de porte empresarial y misión crítica.

ASP.NET versión 4 presenta nuevas características en las áreas siguientes:

- ✓ Servicios principales, incluida una nueva API que permite extender el almacenamiento en caché, la compatibilidad con la compresión de los datos de estado de sesión y un nuevo administrador de precarga de aplicación (característica de inicio automático).

- ✓ Formularios Web Forms, con más compatibilidad integrada con el enrutamiento de ASP.NET, compatibilidad mejorada con estándares web, actualización de la compatibilidad de los exploradores, nuevas características para los controles de datos y nuevas características para la administración de los estados de vista.
  
- ✓ Controles de formularios Web Forms, con un nuevo control Chart.
  
- ✓ MVC, con nuevos métodos de aplicación auxiliar para las vistas, compatibilidad con aplicaciones MVC subdivididas y controladores asincrónicos.
  
- ✓ Datos dinámicos, con compatibilidad con las aplicaciones web existentes, compatibilidad con las relaciones de varios a varios y con la herencia, nuevos atributos y plantillas de campo, y filtrado de datos mejorado.
  
- ✓ Microsoft Ajax, con compatibilidad adicional con las aplicaciones Ajax basadas en cliente en Microsoft Ajax Library.

- ✓ Visual Web Developer, con IntelliSense mejorado para JScript, nuevos fragmentos de código de autocompletar para marcado HTML y ASP.NET, y compatibilidad mejorada con CSS.
  
- ✓ Implementación, con nuevas herramientas para automatizar las tareas de implementación típicas.
  
- ✓ Compatibilidad con múltiples versiones, con mejora del filtrado de las características que no están disponibles en la versión de destino de .NET Framework.

### **2.2.3. Visual studio 2010**

Visual Studio 2010 (VS) es un entorno de desarrollo integrado (IDE), un conjunto de herramientas en una única aplicación que le ayuda a escribir programas. Sin VS, usted tendría que abrir un editor de texto, escribir todo el código, a continuación, ejecutar un compilador de línea de comandos para crear una aplicación ejecutable. El problema con el editor de texto y un compilador de línea de comandos es que se perdería mucho de la productividad a través de procesos manuales.

En las secciones siguientes se explica lo que VS hace.

✓ **Código generado automáticamente.**

VS incluye un conjunto de tipos de proyectos que se pueden elegir. Cada vez que se inicia un nuevo proyecto, VS generará automáticamente el código esqueleto que se puede compilar y ejecutar inmediatamente.

VS ofrece muchos controles prefabricados, los cuales incluyen el código esqueleto, lo que evita tener que escribir su propio código para realizar tareas repetitivas. Muchos de los controles complejos contienen asistentes que le ayudan a personalizar el comportamiento del control.

✓ **Codificación de Rápida Experiencia.**

El editor VS optimiza tu experiencia de codificación. Gran parte de su código está coloreado, tienes el IntelliSense, que te ayuda a medida que vas codificando. VS introduce aún más características, tal como una jerarquía de llamadas, que te permite ver las rutas de llamadas en el código, fragmentos, que permiten escribir una abreviatura que se expande a una plantilla de código, y listas de acciones para generación automática de código nuevo.

✓ **Todo al alcance.**

Usted realmente va a querer aprender cómo navegar en el entorno VS porque una gran cantidad de herramientas están disponibles para ayudarlo en su búsqueda para crear rápidamente software de calidad. Tiene la caja de herramientas repleta de controles, un Explorador de servidores para trabajar con funcionamiento servicios de

sistemas y bases de datos, un explorador de soluciones para trabajar con sus proyectos, pruebas servicios públicos, y los diseñadores visuales.

✓ **Personalización y extensibilidad.**

Puede personalizar muchas partes del entorno de VS, incluyendo colores, opciones de editor, y el diseño.

Las opciones son tan amplias que usted necesita saber dónde buscar para encontrar todos ellos. Si VS no ofrece una función que necesita, puede escribir sus propios macros para automatizar una serie de tareas que se puedan repetir. Para una mayor personalización sofisticada, VS expone una programación de aplicaciones interfaz (API) para la creación de complementos y extensiones.

### **2.2.3.1. Lenguajes**

En Visual Studio 2010, los lenguajes Visual Basic y C# siguen evolucionando hacia la paridad de características. Esto le permite elegir un lenguaje en función de sus preferencias personales, dado que ambos son igualmente capaces. En esta sección se muestran algunas de las nuevas características de C# y Visual Basic.

### **2.2.3.1.1. Lenguaje visual basic**

Visual Basic dispone de nuevas características que acortan la sintaxis y permiten escribir código con más rapidez. Estas características contienen propiedades implementadas automáticamente, continuación de línea implícita, inicializadores de colección y expresiones lambda de una y varias instrucciones. Además, Visual Basic ahora es compatible con la implementación simplificada a través de la equivalencia de tipos.

### **2.2.3.1.2. Lenguaje c#**

Las nuevas características del lenguaje C# permiten simplificar la programación con las API de Microsoft Office. Gracias a los argumentos con nombre y opcionales, ya no es necesario especificar valores para cada parámetro opcional al llamar a métodos COM. La compatibilidad dinámica permite hacer referencia a cualquier objeto, por ejemplo valores devueltos desde API COM o lenguajes dinámicos, sin necesidad de usar una conversión de tipo explícita.

### **2.2.3.1.3. Visual f#**

Visual Studio 2010 incluye F#, un nuevo lenguaje .NET Framework compatible con la programación funcional y la programación

orientada a objetos e imperativa (de procedimientos) convencional. F# combina el estilo sucinto, expresivo y compositivo de la programación funcional con el runtime, las bibliotecas, la interoperabilidad y el modelo de objetos de .NET Framework. En otras palabras, usted se beneficia de lo mejor de ambos paradigmas.

### **2.2.3.2. Acceso a datos y modelado**

.NET Framework 4 Beta 2 incluye mejoras de ADO.NET Entity Framework que reducen aún más la cantidad de código y el mantenimiento necesarios en las aplicaciones controladas por datos. En Visual Studio 2010, la mejora de las opciones de personalización y flexibilidad de las herramientas de ADO.NET Entity Data Model facilitan la creación de aplicaciones de Entity Framework.

### **2.2.3.3. Programación paralela**

Ahora puedes escribir programas que distribuyen trabajo entre varios procesadores sin tener que trabajar directamente con subprocesos ni con el grupo de subprocesos. Visual Studio 2010 incluye bibliotecas de computación paralelas tanto para la biblioteca en tiempo de ejecución de C (CRT) como para .NET Framework. También puede depurar aplicaciones nativas y administradas mediante las nuevas herramientas de multithreading del depurador de Visual Studio.

#### **2.2.4. Modelador de la base de datos Erwin**

Es una herramienta de diseño de base de datos que ayuda a los usuarios a diseñar, generar y mantener alta calidad de las aplicaciones de base de datos de altas prestaciones.

Permite al usuario visualizar la estructura correcta, elementos claves y el diseño optimizado de su base de datos, desde los requerimientos de un modelo lógico de información y reglas de negocio que definen la base de datos, a un modelo físico optimizado para las características específicas de la base de datos seleccionada.

Genera tablas y miles de líneas de procedimientos almacenados y códigos disparadores para las base de datos líderes.

Su tecnología de "comparación completa" permite el desarrollo iterativo, de forma tal que los modelos están siempre sincronizados con la base de datos del usuario.

#### **2.2.5. Base de datos**

Una base de datos es una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales e innecesarias; su finalidad es la de servir a una o más aplicaciones de la mejor manera posible. Los datos se almacenan de modo que resultan independientes de los

programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados.

#### **2.2.5.1. Microsoft Sql server 2008**

Microsoft SQL Server 2008, es un sistema de gestión de bases de datos de datos relacionales, diseñado para procesar grandes volúmenes de información y con la capacidad de cumplir los requerimientos de proceso de información para aplicaciones comerciales y web sites.

Microsoft SQL Server 2008 ofrece el soporte de información para las tradicionales aplicaciones Cliente/Servidor, las cuales están conformadas por una interfaz, a través de la cual, los clientes acceden a los datos por medio de una LAN.

La hoy emergente plataforma .NET exige un gran porcentaje de distribución de recursos, desconexión de los servidores de datos y un entorno de trabajo descentralizado. Para ello, los clientes deben ser livianos, como los browsers de internet, los cuales accederán a los datos por medio de una web Server como el Internet Information Server (IIS).

Microsoft SQL Server 2008 está diseñado para trabajar con dos tipos de datos:

✓ **OLTP (OnLine Transaction Proccesing)**

Son bases de datos caracterizadas por mantener una gran cantidad de usuarios conectados concurrentemente, realizando ingreso/modificación de datos.

✓ **OLAP (OnLine Analytic Processing)**

Son bases de datos que almacenan grandes cantidades de datos que sirven para la toma de decisiones. Por ejemplo: Analizar las ventas en un periodo determinado según la zona de ventas y por cada vendedor.

Los entornos Cliente/Servidor están implementados de tal forma que la información se guarde de forma centralizada en un servidor central. El servidor es responsable de mantener la relación entre los datos, asegurar el correcto almacenamiento de los datos, establecer restricciones que controlen la integridad de datos, etc.

## **2.2.5.2. Ediciones**

### **2.2.5.2.1. Microsoft SQL Server 2008 Enterprise Edition**

Es una edición completa para plataforma de Bussines Intelligence y gestión de datos, esta edición está diseñada para altos niveles de escalabilidad y rendimiento. Use esta edición

para aplicaciones de nivel empresarial en la que los procesos transaccionales en línea (OLTP) son muy exigentes y realizan ciertas tareas de mantenimiento. Esta edición proporciona características de escalabilidad, data warehousing, seguridad, soporte analítico avanzado y reporting de aplicaciones críticas de negocio. Con esta edición, ahora es posible consolidar servidores y ejecutar procesos transaccionales online a gran escala.

#### **2.2.5.2.2. Microsoft SQL Server 2008 Standard Edition**

Standard Edition es la plataforma de análisis y administración de datos para organizaciones pequeñas y medianas. Incluye la funcionalidad básica necesaria para soluciones de línea de negocio, almacenamiento de datos y comercio electrónico.

Las características de alta disponibilidad y de bussiness intelligence integradas de standard edition proporcionan a las organizaciones las capacidades básicas que necesitan para realizar operaciones.

Standard Edition es idóneo para aquella organización pequeña y mediana que necesite

una plataforma completa de análisis y administración de datos.

#### **2.2.5.2.3. Microsoft SQL Server 2008 Workgroup Edition**

Es la solución de administración de datos para pequeñas organizaciones que necesiten una base de datos sin límites de tamaño o número de usuarios.

WorkGroup Edition se puede utilizar como servidor Web de cliente o para operaciones de sucursales o departamentos. Incluye las características de base de datos principales de la línea de productos de SQL Server y es fácilmente actualizable a Standard o Enterprise Edition.

WorkGroup Edition es una base de datos de nivel de entrada idónea, que resulta confiable, sólida y fácil de administrar.

#### **2.2.5.2.4. Microsoft SQL Server 2008 Web Edition**

Web Edition esta específicamente diseñado para entornos web de alta disponibilidad que se ejecutan sobre Windows Server. SQL Server 2008 Web proporciona lo necesario para soportar aplicaciones web altamente escalables.

#### **2.2.5.2.5. Microsoft SQL Server 2008 Developer Edition**

Developer Edition permite a los desarrolladores construir y probar cualquier tipo de aplicaciones con SQL Server 2008. Esta edición incluye todas las funcionalidades de SQL Server Enterprise, pero contiene licencia solo para desarrollar, probar y utilizar como demo. Las aplicaciones y las bases de datos desarrolladas bajo esta edición pueden ser actualizadas fácilmente SQL Server Enterprise.

#### **2.2.5.2.6. Microsoft SQL Server 2008 Express Edition**

Express Edition es una versión gratuita de SQL Server que incluye funcionalidades claves de base de datos, entre las que se le incluyen todos los tipos de datos del nuevo SQL Server 2008 ahora en una aplicación ligera. Esta edición es perfecta para aprender y construir escritorios y pequeñas aplicaciones de servidor, y para ser redistribuidas como aplicaciones clientes.

#### **2.2.5.2.7. Microsoft SQL Server 2008 Compact Edition**

Compact Edition es una base de datos embebida gratuita diseñada para desarrolladores, idónea para construir aplicaciones independientes y ocasionalmente conectadas para dispositivos móviles, equipos

portátiles, desktops y clientes Web. SQL Server Compact funciona sobre todas las plataformas Microsoft Windows, incluyendo los sistemas operativos Windows XP y Windows Vista, y en dispositivos Pocket Pc y Smartphone.

### 2.2.6. Windows 7

Windows 7 es la mejor versión de Windows nunca antes vista. Tal vez parte de la buena acogida se debe a su contraste con Windows Vista, que fue casi universalmente despreciado. Técnicamente es una evolución de Windows Vista, por lo que Windows 7 mantiene todo lo bueno que tenía de Windows Vista: la estabilidad, la seguridad, la animación sólo lo suficiente, para mantener las cosas interesantes.

Aún Windows 7 mantiene lo que todos odiaban de vista:

- ✓ **Velocidad.** En las pruebas de PC Magazine, Windows 7 fue de 12 a 14 por ciento más rápido que Vista. Es especialmente rápido en el arranque, en suspender y al reanudar.
- ✓ **Requerimientos de hardware.** Los ordenadores han crecido más rápido y son más potentes desde el debut de Windows Vista en 2007, pero el requerimiento de hardware para Windows 7 es exactamente lo mismo. Incluso las notebooks de \$ 300 logran ejecutar Windows 7.
- ✓ **Seguridad.** Windows vista utiliza pantallas completas para cuadros de advertencia a cada amenaza

potencial para la seguridad. Pero Windows 7 te deja en paz mucho más a menudo. De hecho, 10 categorías de advertencias ahora se acumulan en silencio en un panel de control unificado simple, llamado el Centro de Acción, y no te interrumpe para nada.

### **2.2.7. Windows server 2008**

Windows Server 2008 es el nombre de un sistema operativo de Microsoft diseñado para servidores.

Es el sucesor de Windows Server 2003, distribuido al público casi cinco años después. Al igual que Windows 7, Windows Server 2008 se basa en el núcleo Windows NT 6.1. . Entre las mejoras de esta edición, se destacan nuevas funcionalidades para el Active Directory, nuevas prestaciones de virtualización y administración de sistemas, la inclusión de IIS 7.5 y el soporte para más de 256 procesadores.

Hay algunas diferencias (unas sutiles y otras no tanto) con respecto a la arquitectura del nuevo Windows Server 2008, que pueden cambiar drásticamente la manera en que se usa este sistema operativo. Estos cambios afectan a la manera en que se gestiona el sistema hasta el punto de que se puede llegar a controlar el hardware de forma más efectiva, se puede controlar mucho mejor de forma remota y cambiar de forma radical la política de seguridad. Entre las mejoras que se incluyen, están:

- ✓ Nuevo proceso de reparación de sistemas NTFS: proceso en segundo plano que repara los archivos dañados.

- ✓ Creación de sesiones de usuario en paralelo: reduce tiempos de espera en los Terminal Services y en la creación de sesiones de usuario a gran escala.
- ✓ Cierre limpio de Servicios.
- ✓ Sistema de archivos SMB2: de 30 a 40 veces más rápido el acceso a los servidores multimedia.
- ✓ Address Space Load Randomization (ASLR): protección contra malware en la carga de controladores en memoria.
- ✓ Windows Hardware Error Architecture (WHEA): protocolo mejorado y estandarizado de reporte de errores.
- ✓ Virtualización de Windows Server: mejoras en el rendimiento de la virtualización.
- ✓ PowerShell: inclusión de una consola mejorada con soporte GUI para administración.
- ✓ Server Core: el núcleo del sistema se ha renovado con muchas y nuevas mejoras.

### **2.2.8. JQuery**

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML,

manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

#### **2.2.8.1. Características**

- ✓ Selección de elementos DOM.
- ✓ Interactividad y modificaciones del árbol DOM, incluyendo
- ✓ soporte para CSS 1-3 y un plugin básico de XPath.
- ✓ Manipulación de la hoja de estilos CSS.
- ✓ Efectos y animaciones.
- ✓ Animaciones personalizadas.
- ✓ AJAX.

- ✓ Soporta extensiones.
- ✓ Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- ✓ Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.5

#### **2.2.8.2. Uso**

jQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Para ello utiliza las funciones `$()` o `jQuery()`.

#### **2.2.9. GanttProject**

GanttProject es una aplicación que permite organizar y planificar proyectos a través de diagramas Gannt, como así también la carga horaria de los integrantes dentro del mismo. A diferencia de Visual Project de Microsoft, GanntProject es una aplicación open source, por lo que se puede descargar de forma gratuita y además es una aplicación íntegramente escrita en Java por lo que lo hace multiplataforma.

##### **2.2.9.1. Características**

- ✓ Generación de Diagramas PERT.
- ✓ Reportes en PDF o HTML.
- ✓ Diagrama de Gantt.
- ✓ Posibilidad de importar y exportar archivos de MS Project.
- ✓ Organización de tareas en forma jerárquica.

#### **2.2.9.2. Ventajas de GanttProject**

- ✓ Se puede descargar de forma gratuita y además es una aplicación íntegramente escrita en Java por lo que lo hace multiplataforma.
- ✓ Reportes en PDF o HTML.
- ✓ Posibilidad de importar y exportar archivos de MS Project.
- ✓ Organización de tareas en forma jerárquica.
- ✓ Permite crear diagramas y asignares tiempos, recursos.
- ✓ Otra variante que se puede utilizar en combinación, es el diagrama de Pert.
- ✓ Fácil aprendizaje.
- ✓ Precio excelente.
- ✓ Las herramientas de gestión de proyectos comerciales cuestan una gran cantidad de dinero.
- ✓ Gratuito para cualquier propósito.
- ✓ Interfaz amigable

### **2.2.9.3. Microsoft reporting technologies**

Los reportes son componentes importantes de las aplicaciones y son usadas principalmente para proveer una vista descriptiva y resumida de datos la cual permite una correcta toma de decisiones.

Tambien llamado Reporting Services. La arquitectura de cliente de Reporting Services se basa en el componente del lado del cliente (el ReportViewer) que recoge los datos de la fuente de datos y los procesa para producir y presentar los informes a los usuarios.

El Data Source es un componente importante para desarrollar informes. El DataSet de ADO .NET es la interfaz de datos proporcionado por Visual Studio y altamente integrado con Reporting Services (del lado del cliente y del lado del servidor). La definición del informe se produce utilizando el Report Definition Language (RDL), que es un archivo basado en XML.

La arquitectura de servidor de Reporting Services es un componente basado en un servidor llamado Reporting Server que almacena la definición de los informes, como Data Source y parámetros. Cuando el usuario final necesita acceder a los reportes usando ReportViewer, este se conecta al Reporting Server y la solicitud de informe de destino. Reporting Server se conecta a la fuente de datos, recopila los datos y

enlaza los datos a la definición del informe (también almacenado en el Reporting Server ) y finalmente envía el informe de salida al ReportViewer que presenta el informe al usuario final. Reporting Server controla la autenticación y autorización de acceso de los usuarios a los informes.

## **2.2.10. Tecnologías adicionales para el desarrollo**

### **2.2.10.1. Html5**

HTML5 hace de la Web un entorno de primera clase para la creación de aplicaciones reales.

HTML5 refuerza es un conjunto de herramientas existentes de JavaScript con extensiones claves para las API del navegador que facilitan la creación de aplicaciones. La web comenzó como una forma de compartir archivos, almacenados en un servidor web.

Desarrolladores rápidamente encontraron la manera de generar esos archivos sobre la marcha, dando el primer gran paso hacia la creación de aplicaciones. El siguiente gran paso fue la adición de la interactividad en el cliente de navegador. JavaScript y el Document Object Model (DOM) permiten a los desarrolladores crear HTML dinámico. Después de algunos años, el Ajax trajo estas técnicas a ponerse de moda, agregando algunas herramientas para dejar que las páginas se comunican con el servidor en trozos más pequeños. HTML5 en estos veinte años de desarrollo,

ha llenado algunos vacíos críticos. Muchos de los cambios de HTML5 soportan nuevas funciones (especialmente multimedia y gráficos) que habían requerido previamente plugins, pero en el fondo, da a los programadores de JavaScript las herramientas que necesitan para crear de forma independiente (o por lo menos de manera más flexible) las aplicaciones que utilizan HTML para la estructura, CSS para la presentación, y JavaScript para la lógica y el comportamiento.

### **2.2.10.2. Css3**

El objetivo inicial de CSS, separar el contenido de la forma, se cumplió ya con las primeras especificaciones del lenguaje. Sin embargo, el objetivo de ofrecer un control total a los diseñadores sobre los elementos de la página ha sido más difícil de cubrir. Las especificaciones anteriores del lenguaje tenían muchas utilidades para aplicar estilos a las webs, pero los desarrolladores aún continúan usando trucos diversos para conseguir efectos tan comunes o tan deseados como los bordes redondeados o el sombreado de elementos en la página.

CSS 1 ya significó un avance considerable a la hora de diseñar páginas web, aportando mucho mayor control de los elementos de la página. Pero como todavía quedaron muchas otras cosas que los diseñadores deseaban hacer, pero que CSS no permitía especificar, éstos debían hacer uso de trucos

para el diseño. Lo peor de esos trucos es que muchas veces implica alterar el contenido de la página para incorporar nuevas etiquetas HTML que permitan aplicar estilos de una manera más elaborada. Dada la necesidad de cambiar el contenido, para alterar al diseño y hacer cosas que CSS no permitía, se estaba dando al traste con alguno de los objetivos para los que CSS fue creado, que era el separar por completo el contenido de la forma.

CSS 2 incorporó algunas novedades interesantes, que hoy ya utilizamos habitualmente, pero CSS 3 todavía avanza un poco más en la dirección, de aportar más control sobre los elementos de la página.

Así pues, la novedad más importante que aporta CSS 3, de cara a los desarrolladores de webs, consiste en la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos o hacks, que a menudo complicaban el código de las web.

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a:

- ✓ Un web entero, de modo que se puede definir la forma de todo el web de una sola vez.

- ✓ Un documento HTML o página, se puede definir la forma, en un pequeño trozo de código en la cabecera, a toda la página.
- ✓ Una porción del documento, aplicando estilos visibles en un trozo de la página.
- ✓ Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece potencia en nuestra programación. Podemos definir, por ejemplo, varios tipos de párrafos: en rojo, en azul, con márgenes y sin ellos.

Esta tecnología es bastante nueva, por lo que no todos los navegadores la soportan. En concreto, sólo los navegadores de Netscape versiones de la 4 en adelante y de Microsoft a partir de la versión 3 son capaces de comprender los estilos en sintaxis CSS. Además cabe destacar que no todos los navegadores implementan las mismas funciones de hojas de estilos, por ejemplo, Microsoft Internet Explorer 3 no soporta todo lo relativo a capas.

### **2.2.10.3. Ajax**

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se

ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Ajax es una combinación de cuatro tecnologías ya existentes:

- ✓ XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- ✓ Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- ✓ El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios. PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico también utilizado en el método Ajax.
- ✓ XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

## **2.2.11. Servicio web**

### **2.2.11.1. Internet information server 7**

IIS es un conjunto de servicios para servidores usando Microsoft Windows. Es especialmente usado en servidores web, que actualmente es el segundo más popular sistema de servidor web (funciona en el 35% de los servidores de todos los sitios web).

De hecho, el IIS viene integrado con Windows NT 4.0. Dado que el IIS está tan íntimamente integrado con el sistema operativo, es relativamente fácil de administrar. Sin embargo, actualmente el IIS está disponible sólo para el uso en la plataforma Windows NT, mientras que los servidores Web de Netscape corren en la mayoría de las plataformas, incluyendo Windows NT, OS/2 y UNIX.

## **2.2.12. Red de datos**

Se denomina red de datos a aquellas infraestructuras o redes de comunicación que se ha diseñado específicamente a la transmisión de información mediante el intercambio de datos.

Las redes de datos se diseñan y construyen en arquitecturas que pretenden servir a sus objetivos de uso. Las redes de datos, generalmente, están basadas en la conmutación de paquetes y se clasifican de acuerdo a su tamaño, la distancia que cubre y su arquitectura física.

### **2.2.12.1. Clases de redes de datos.**

**2.2.12.1.1. Red de Área Local (LAN):** Las redes de área local suelen ser una red limitada la conexión de

equipos dentro de un único edificio, oficina o campus, la mayoría son de propiedad privada.

**2.2.12.1.2. Red de Área Metropolitana (MAN):** Las redes de área metropolitanas están diseñadas para la conexión de equipos a lo largo de una ciudad entera. Una red MAN puede ser una única red que interconecte varias redes de área local LAN's resultando en una red mayor. Por ello, una MAN puede ser propiedad exclusivamente de una misma compañía privada, o puede ser una red de servicio público que conecte redes públicas y privadas.

**2.2.12.1.3. Red de Área Extensa (WAN):** Las Redes de área extensa son aquellas que proporcionen un medio de transmisión a lo largo de grandes extensiones geográficas (regional, nacional e incluso internacional). Una red WAN generalmente utiliza redes de servicio público y redes privadas y que pueden extenderse alrededor del globo.

## **2.2.12.2. Topología**

Se denomina topología a la forma geométrica en la que se encuentran distribuidos las estaciones de trabajo y los cables que las conectan.

Las estaciones de trabajo de una red se comunican entre sí mediante una conexión física y el objeto de la topología es el de buscar la forma más económica y eficaz de conectarlas para, al mismo tiempo, facilitar la fiabilidad del sistema, evitar los tiempos de espera

en la transmisión de los datos, permitir un mejor control de la red y permitir de forma eficiente el aumento de las estaciones de trabajo. Las formas más utilizadas son:

**2.2.12.2.1. Configuración en bus:** En ella todas las estaciones comparten el mismo canal de comunicaciones, toda la información circula por ese canal y cada una de ellas recoge la información que le corresponde.

Esta configuración es fácil de instalar, la cantidad de cable a utilizar es mínima, tiene una gran flexibilidad a la hora de aumentar o disminuir el número de estaciones y el fallo de una estación no repercute en la red, debido a que los datos pasan de largo por las estaciones de trabajo, en vez de pasar a través de ellos, aunque la ruptura de un cable dejará la red totalmente inutilizada. Las estaciones de trabajo, a su vez actúan de concentradores o hubs. La mayor desventaja de esta configuración es que debido a los pocos puntos de concentración, son difíciles el diagnóstico y el aislamiento de fallos.

**2.2.12.2.2. Configuración en anillo:** En ella todas las estaciones están conectadas entre sí formando un anillo, de forma que cada una sólo tiene contacto directo con otras dos.

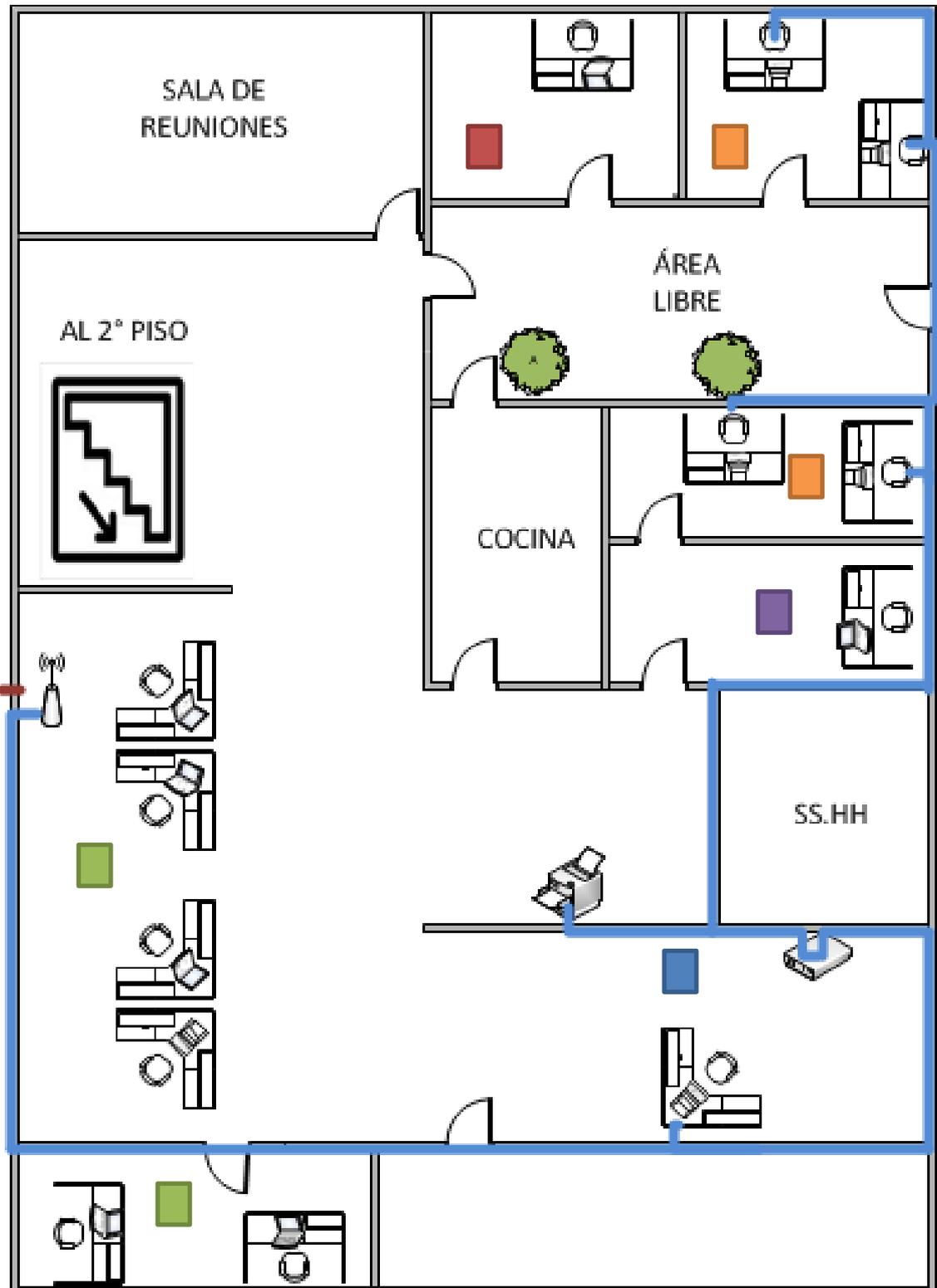
Este tipo de redes permite aumentar o disminuir el número de estaciones sin dificultad, pero, a medida que aumenta el flujo de información, será menor la velocidad de respuesta de la red.

**2.2.12.2.3. Configuración en estrella:** Esta forma es una de las más antiguas y, en ella, todas las estaciones están conectadas directamente al servidor o a un ordenador central y todas las comunicaciones se han de hacer necesariamente a través de él. Permite incrementar y disminuir fácilmente el número de estaciones.

## 2.2.13. Red e internet de AIF contratistas generales S.A.C

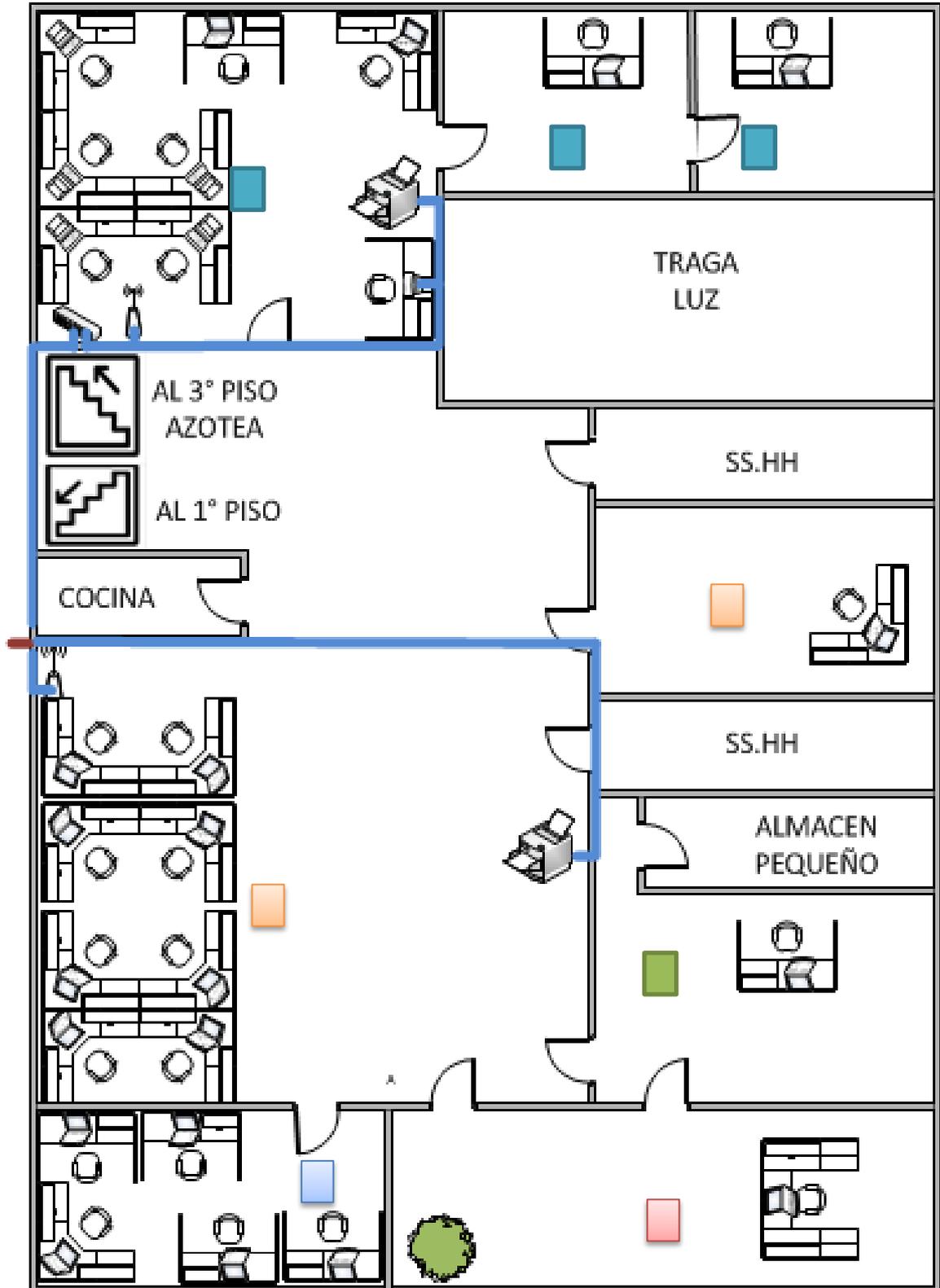
### 2.2.13.1. Primer piso:

Img.10: Estructura de Red primer piso



### 2.2.13.2. Segundo piso

Img.11: Estructura de Red segundo piso



### 2.2.13.3. Leyenda

#### 2.2.13.3.1. Estructura y equipos:

-  Cableado de red categoría 5.
-  Conexión de red entre el 1er y 2do piso.
-  Computadora de escritorio.
-  Laptop.
-  Impresora/Fotocopiadora/escáner.
-  Access point.
-  Modem Router.
-  Switch de 10 puertos.

#### 2.2.13.3.2. Áreas:

-  Recepción.
-  Logística.
-  Ventas.
-  Contabilidad
-  Construcción.
-  Arquitectura.
-  Administración.
-  Auditoria.
-  Gerente.

## **2.2.14. Internet**

### **2.2.14.1. 1 MB:**

#### **2.2.14.1.1. Características:**

- Acceso dedicado Simétrico.
- Velocidad de transmisión de datos de: 64 Kbps a 100 Mbps.
- Brinda 16 direcciones IP públicas permanentes, escalable de acuerdo a las necesidades del cliente.
- El servicio incluye uso de Router Cisco.
- Soporte Online las 24 horas del día, los 365 días del año llamando al 1366 (Número exclusivo de empresas).
- Sistema de monitoreo permanente sobre la velocidad del servicio contratado.

#### **2.2.14.1.2. Aplicaciones:**

- Ideal para el acceso e implementación de servicios de Internet como Páginas Web (www), E-mail (SMTP), transferencia de archivos (FTP), simulación de terminales (TELNET), Dial-Pad, IRC (Comunicación en tiempo real), videoconferencia, comercio electrónico (E-business), entre otros.
- Alojamiento de Dominios en nuestros DNS (sistema de nombres de dominios).

### **3. CAPITULO III**

## **PLANIFICACIÓN DEL PROYECTO**

### **3.1. Planificación del proyecto**

El objetivo principal del Sistema de Gestión de Tramite Documentario es llevar un control y seguimiento detallado de todos los expedientes que se procesan en la institución, tanto internos como externos y permitir a la institución tener la ubicación física y lógica de la documentación que llega y fluye dentro de ella, así como de la que se genera al interior de la misma.

#### **3.1.1. Alcance del sistema**

##### **✓ Mantenimiento**

- Gestión de áreas.
- Gestión de usuarios.
- Gestión de Procesos.
- Gestión de contraseña.
- Gestión de año.
- Gestión de parámetros.

##### **✓ Tramite**

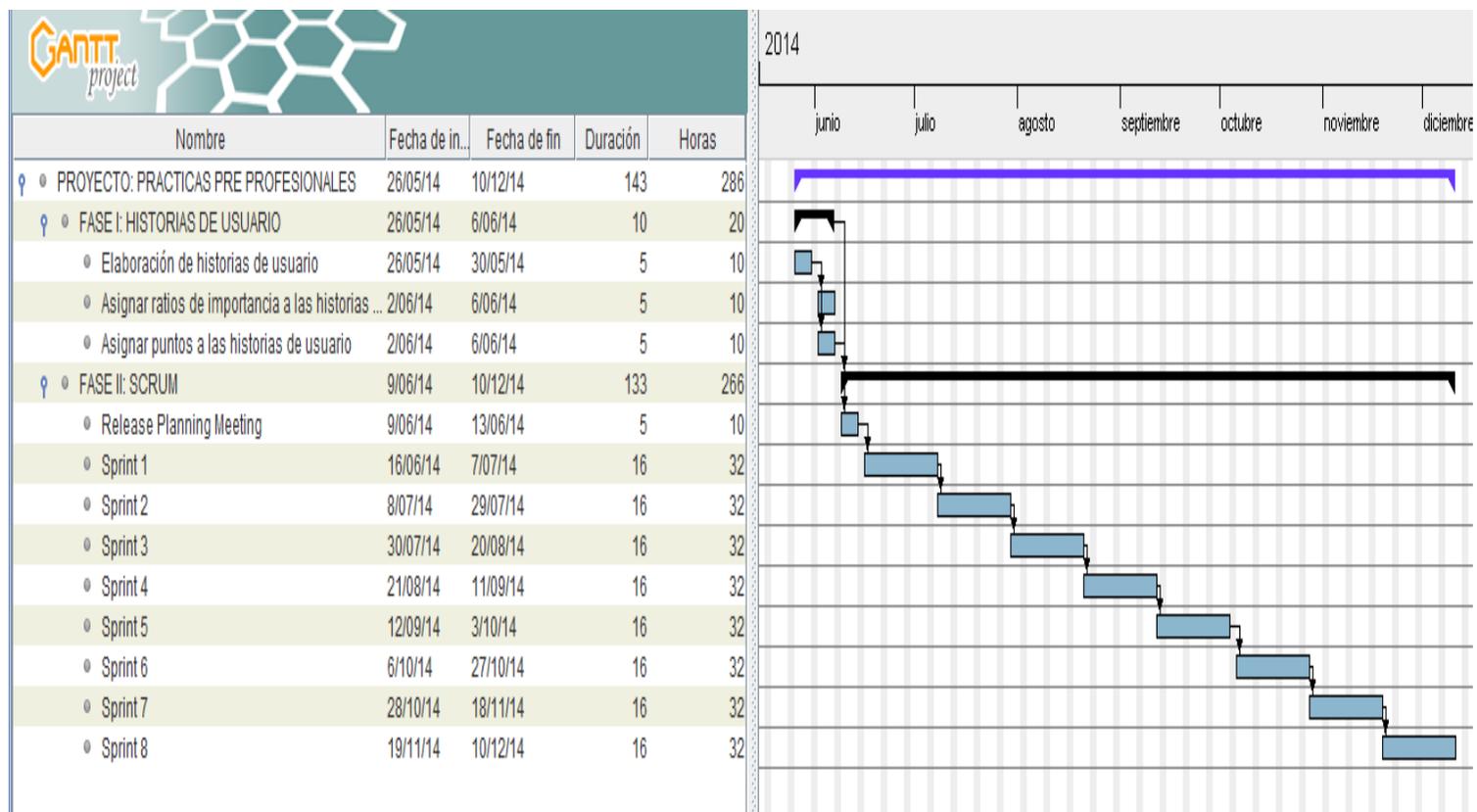
- Inicio de Sesión.
- Interfaz Principal.
- Registrar Captado.
- Registrar Beneficiario.
- Generar Documentos.
- Modificar Beneficiario.
- Control por Beneficiario.

##### **✓ Consultas**

- Consultar Captados.
- Consultar Beneficiarios.
- Consultar Obras.
- Consultar Registros.

### 3.1.2. Cronograma de actividades del proyecto

Img.12: Cronograma de Actividades del Proyecto



### **3.1.3. Factores críticos de éxito**

- ✓ Investigar sobre sistemas implantados con anterioridad.
- ✓ Definir las iteraciones y las funcionalidades que formarán parte del alcance del sistema a implementar.
- ✓ Definir los estándares de desarrollo y documentación necesaria para el inicio de la ejecución del proyecto.
- ✓ Interés de parte de la alta dirección de la empresa y de parte del equipo de desarrollo.

### **3.1.4. Supuestos**

- ✓ El equipo de desarrollo deberá capacitar a los usuarios finales del sistema.
- ✓ La empresa constructora AIF CONTRATISTAS GENERALES S.A.C deberá adquirir un servidor y dominio para alojar la aplicación web.

### **3.1.5. Restricciones**

- ✓ El sistema involucrará los módulos de registro, consultas y gestión.
- ✓ Los usuarios que se autentiquen en el sistema solo tendrán acceso a ciertas operaciones, según los roles y permisos asignados.

### 3.1.6. Análisis de factibilidad

#### 3.1.6.1. Factibilidad técnica

Tbl.7: Factibilidad técnica Software

SOFTWARE	
Lo que se requiere	Lo que se tiene
<b>Sistemas Operativos:</b> <ul style="list-style-type: none"> <li>Windows Server 2008.</li> <li>Windows 7 o posterior.</li> </ul>	<b>Sistemas Operativos:</b> <ul style="list-style-type: none"> <li>Windows 7.</li> <li>Winsdows 8.</li> </ul>
<b>Utilitarios:</b> <ul style="list-style-type: none"> <li>Microsoft office 2010 o superior.</li> </ul>	<b>Utilitarios:</b> Microsoft office 2010.
<b>Navegador:</b> <ul style="list-style-type: none"> <li>Chrome.</li> </ul>	<b>Navegador:</b> <ul style="list-style-type: none"> <li>Chrome.</li> </ul>
<b>SGBD:</b> SQL Server 2008.	
<b>Entorno de desarrollo:</b> Visual Studio 2010.	

Tbl.8: Factibilidad técnica Hardware

HARDWARE	
Lo que se requiere	Lo que se tiene
<b>PC-SERVIDOR:</b> requerimientos mínimos <ul style="list-style-type: none"> <li>CPU Intel Core I5.</li> <li>6 Gb de RAM.</li> <li>4.0 GHz.</li> <li>Disco duro de 500 Gb.</li> <li>Lectora Dvd-Rom.</li> </ul>	
<b>Cliente:</b> requerimientos mínimos. <ul style="list-style-type: none"> <li>CPU Dual Core.</li> <li>2 Gb de RAM.</li> <li>2.0 GHz.</li> <li>Disco duro de 80 Gb.</li> </ul>	<b>Cliente:</b> <ul style="list-style-type: none"> <li>Intel Core i5</li> <li>4Gb de RAM.</li> <li>3.0 GHz.</li> <li>Disco duro de 500 Gb.</li> </ul>

### 3.1.6.2. Factibilidad operativa

- El sistema permitirá llevar un control sobre los procesos documentarios que se realizan, permitirá la centralización de la data actual.
- El sistema que se va a implantar en dicha empresa va a cumplir con los requerimientos que solicite el usuario; además se brindara la capacitación respectiva a los usuarios del sistema.

### 3.1.6.3. Factibilidad económica

Tbl.9: Factibilidad económica Software

<b>SOFTWARE</b>	
<b>Item</b>	<b>Costo</b>
Windows Server 2008	S/.1200.00
Windows 7	S/.250.00
SQL Server 2008	S/.450.00
Visual Studio 2010	S/.320.00
Microsoft Office 2010	S/.250.00
Antivirus	S/.320.00
Navegadores	S/,0.00
<b>TOTAL</b>	<b>S/.2790.00</b>

Tbl.10: Factibilidad económica Hardware

<b>HARDWARE</b>	
<b>Item</b>	<b>Costo</b>
Servidor	S/.3500.00
<b>TOTAL</b>	<b>S/.3500.00</b>

Tbl.11: Factibilidad económica Recursos Humanos

<b>RECURSOS HUMANOS</b>		
<b>Personal</b>	<b>Meses</b>	<b>Total</b>
Analista	1	S/.1200.00
Diseñador	1	S/.1100.00
Programador	3	S/.2850.00
Prog. BD	1	S/.1700.00
<b>TOTAL</b>		<b>S/.6850.00</b>

Tbl.12: Factibilidad económica Flujo de pago

<b>FLUJO DE PAGO</b>	
<b>Recursos</b>	<b>Costo</b>
Software	S/.2790.00
Hardware	S/.3500.00
Recursos humanos	S/.6850.00
Imprevistos (10%)	S/.1314.00
<b>TOTAL</b>	<b>S/.14450.00</b>

## **4. CAPITULO IV**

### **DESARROLLO DEL SISTEMA**

## 4.1. Análisis del sistema.

### 4.1.1. Definición de estándares.

#### 4.1.1.1. Estándares para las historias de usuario o Pila del Producto.

Los estándares descritos a continuación están referidos a las fichas donde se describirán las historias de usuario a implementar en cada iteración:

- ✓ **ID:** un identificador único, simplemente un número auto-incremental. Esto nos permite no perder la pista a las historias cuando cambiamos su nombre.
- ✓ **Nombre:** una descripción corta de la historia. Por ejemplo, "Ver tu historial de transacciones". Suficientemente claro como para que el dueño de producto comprenda aproximadamente de qué estamos hablando, y suficientemente clara como para distinguirla de las otras historias. Normalmente, 2 a 10 palabras.
- ✓ **Importancia:** el ratio de importancia que el Dueño de producto da a esta historia. Por ejemplo, 10. O 150. Más alto = más importante.
- ✓ **Estimación inicial:** la valoración inicial del Equipo acerca de cuanto trabajo es necesario para implementar la historia, comparada con otras historias. La unidad son "puntos de historia" y usualmente corresponde a "días-persona ideales". Pregunta al Equipo: "si tuvierais el número óptimo de personas para esta historia (ni muchos

ni pocos, típicamente 2) y os encerraseis en una habitación con cantidad de comida, y trabajaseis sin distracciones, ¿en cuántos días saldríais con una implementación terminada, demostrable, testeada y liberable?". Si la respuesta es "con 3 personas encerrados en una habitación nos llevaría 4 días", entonces la estimación inicial son 12 puntos. Lo importante no es que las estimaciones absolutas sean correctas (es decir, que una historia de 2 puntos deba durar 2 días), lo importante es que las estimaciones relativas sean correctas (es decir, que una historia de 2 puntos debería durar la mitad que una historia de 4 puntos).

- ✓ **Como probarlo:** una descripción a alto nivel de cómo se demostrará esta historia en la Demo al final del Sprint. Se trata, esencialmente, de una simple especificación de un test: "Haz esto, entonces haz lo otro, y entonces debería ocurrir aquello".
- ✓ **Notas:** cualquier otra información, clarificación, referencia a otras fuentes de información, etc. Normalmente muy breve.

#### 4.1.1.2. Estándares para el Plan de Entrega.

Ejemplo de umbrales de aceptación:

- ✓ Todos los elementos con importancia  $\geq 100$  deben estar incluidos en la versión 1.0, o se penalizara al equipo de trabajo.

- ✓ Todos los elementos de importancia 50-99 deberían estar incluidos en la versión 1.0, pero podríamos pasar sin ellos si los incluyésemos en otra entrega poco después.
- ✓ Los elementos con importancias 25-49 son requisitos, pero podemos incluirlos en una versión 1.1.
- ✓ Los elementos con importancia <25 son puramente especulativos y puede que ni siquiera hagan falta.

Ejemplo de Pila de Producto con un código de colores basado en las reglas anteriores (en puntos de historia):

Img.13: Ejemplo de Pila de producto

Importancia	Historia
130	Plátano
120	Manzana
115	Naranja
110	Guayaba
100	Pera
95	Pasa
80	Cacahuete
70	Donut
60	Cebolla
40	Uva
35	Papaya
10	Arándano
10	Melocotón

Rojo = debe incluirse en la versión 1.0 (plátano – pera)

Amarillo = debería incluirse en la versión 1.0 (pasa – cebolla)

Verde = puede hacerse más tarde (uva – melocotón)

Si el tiempo se acaba, se podría salir adelante abandonando pasa, cacahuete, donut o cebolla. Cualquier cosa por debajo de cebolla es un plus.

## 4.2. Historias de usuario

Tbl.13: Historia de Usuario Fabiola Vasquez

<i>Proyecto: SISTEMATIZACION DE PROCESOS PARA LA GESTION DE LA INFORMACION DOCUMENTARIA ENTRE LAS AREAS DE ADMINISTRACION Y CONSTRUCCION DE LA EMPRESA CONSTRUCTORA "AIF CONSTRATISTAS GENERALES S.A.C"</i>			
<i>CHINCHA ALTA – 2014</i>			
26/05/2014	AUTOR: FABIOLA VASQUEZ	CARGO: ASISTENTE	AREA:ADMINISTRACION

<b>HISTORIAS DE USUARIO</b>
-----------------------------

ID	NOMBRE	ESTIMACION	COMO PROBARLO	NOTAS
AD001	REGISTRAR DATOS DE CAPTADOS MEDIANTE PAGINA WEB	10	MOSTRAR EN BUSQUEDA LOS CAPTADOS POR CAPTADOR	CLARIFICAR REQUISITOS PARA LA CREACION DE LA BD
AD002	REGISTRAR DATOS DE BENEFICIARIO ELEGIBLE MEDIANTE PAGINA WEB	24	MOSTRANDO INFORMACION DE BENEFICIARIO EN UNA BUSQUEDA	CLARIFICAR REQUISITOS PARA LA CREACION DE LA BD
AD003	GENERAR NUMERO DE CONTRATO Y MOSTRAR EN PAG WEB	8	VISUALISAR EN PANTALLA EL NUMERO DE CONTRATO GENERADO	
AD004	ACTUALIZAR DATOS PARA REGISTRO	10	MUESTRA DE LOS DATOS MODIFICADOS DEL BENEFICIARIO	

Tbl.14: Historia de Usuario Magaly Siguas

*Proyecto: SISTEMATIZACION DE PROCESOS PARA LA GESTION DE LA INFORMACION DOCUMENTARIA ENTRE LAS AREAS DE ADMINISTRACION Y CONSTRUCCION DE LA EMPRESA CONSTRUCTORA "AIF CONSTRATISTAS GENERALES S.A.C" CHINCHA ALTA – 2014*

26/05/2014	Autor: MAGALY SIGUAS	CARGO: ASISTENTE	AREA:ADMINISTRACION
------------	----------------------	------------------	---------------------

### HISTORIAS DE USUARIO

ID	NOMBRE	ESTIMACION	COMO PROBARLO	NOTAS
AD005	QUE EL CAPTADOR CARGUE ESCANEOS DE DOCUMENTOS BASE DEL BENEFICIARIO AL SERVIDOR	16	QUE EL INGRESO DE CADA DOCUMENTO SEA INDEPENDIENTE,QUE SE PUEDA VISUALIZAR ARCHIVOS EN EL SERVIDOR	CLARIFICAR MOMENTO DE CARGA
AD006	ACTUALIZAR ESTADO PRE FMV	8	MOSTRAR DATOS MODIFICADOS	
AD007	ACTUALIZAR ESTADO POST ENVIO FMV	8	MOSTRAR DATOS MODIFICADOS	

Tbl.15: Historia de Usuario Fiorella Perez

<i>Proyecto: SISTEMATIZACION DE PROCESOS PARA LA GESTION DE LA INFORMACION DOCUMENTARIA ENTRE LAS AREAS DE ADMINISTRACION Y CONSTRUCCION DE LA EMPRESA CONSTRUCTORA "AIF CONSTRATISTAS GENERALES S.A.C" CHINCHA ALTA – 2014</i>			
26/05/2014	AUTOR: FIORELLA PEREZ	CARGO: ASISTENTE	AREA:ADMINISTRACION

<b>HISTORIAS DE USUARIO</b>
-----------------------------

ID	NOMBRE	ESTIMACION	COMO PROBARLO	NOTAS
AD008	INGRESAR CODIGO FMV AL BENEFICIARIO REGISTRADO	16	LUEGO DE INGRESAR CODIGO MOSTRAR DATO DEL BENEFICIARIO CORRESPONDIENTE	
AD009	INGRESO DE AHORRO POR PAGINA WEB	8	LUEGO DE INGRESAR AHORRO MOSTRAR DATO DEL BENEFICIARIO CORRESPONDIENTE	

Tbl.16: Historia de Usuario Juan Carlos de la Cruz

<i>Proyecto: SISTEMATIZACION DE PROCESOS PARA LA GESTION DE LA INFORMACION DOCUMENTARIA ENTRE LAS AREAS DE ADMINISTRACION Y CONSTRUCCION DE LA EMPRESA CONSTRUCTORA "AIF CONSTRATISTAS GENERALES S.A.C"</i> <i>CHINCHA ALTA – 2014</i>			
26/05/2014	AUTOR: JUAN CARLOS DE LA CRUZ	CARGO: ASISTENTE	AREA: ADMINISTRACION-ASIGNACION

<b>HISTORIAS DE USUARIO</b>
-----------------------------

ID	NOMBRE	ESTIMACION	COMO PROBARLO	NOTAS
AS001	ASIGNACION A CARTA FIANZA	12	MOSTRAR CF CON BENEFICIARIOS CORRESPONDIENTES	

Tbl.17: Historia de Usuario Roberto Soto

<i>Proyecto: SISTEMATIZACION DE PROCESOS PARA LA GESTION DE LA INFORMACION DOCUMENTARIA ENTRE LAS AREAS DE ADMINISTRACION Y CONSTRUCCION DE LA EMPRESA CONSTRUCTORA "AIF CONSTRATISTAS GENERALES S.A.C" CHINCHA ALTA – 2014</i>			
26/05/2014	AUTOR: ROBERTO SOTO	CARGO: ASISTENTE	AREA:CONSTRUCCION

<b>HISTORIAS DE USUARIO</b>
-----------------------------

ID	NOMBRE	ESTIMACION	COMO PROBARLO	NOTAS
CO001	ACTUALIZAR ESTADO DE OBRA	6	MOSTRAR LA MODIFICACION DEL ESTADO EN UNA TABLA	
CO002	IMPRIMIR DOCUMENTOS BASE DE BENEFICIARIO	12	IMPRESION DE DOCUMENTOS CORRESPONDIENTES A UN BENEFICIARIO	

Tbl.18: Historia de Usuario Ing. Marco Herrera

<i>Proyecto: SISTEMATIZACION DE PROCESOS PARA LA GESTION DE LA INFORMACION DOCUMENTARIA ENTRE LAS AREAS DE ADMINISTRACION Y CONSTRUCCION DE LA EMPRESA CONSTRUCTORA "AIF CONSTRATISTAS GENERALES S.A.C"</i>			
<i>CHINCHA ALTA – 2014</i>			
26/05/2014	AUTOR: ING.MARCO HERRERA	CARGO: GERENTE	AREA:GERENCIA

### HISTORIAS DE USUARIO

ID	NOMBRE	ESTIMACION	COMO PROBARLO	NOTAS
GE001	CUADRO DE PLANIFICACION	15	VISUALIZAR EN EL CUADRO LAS LIBERACIONES PLANIFICADAS	CLARIFICAR SI LO PLANIFICADO SE INGRESA O SE OBTIENE DE FECHAS DE LIBERACION DE CF
GE002	CUADRO DE EJECUCION	12	VISUALIZAR EN CUADRO LIBERACIONES CONCLUIDAS	
GE003	CUADRO DE DIFERENCIA	10	VISUALIZAR EN EL CUADRO LA DIFERENCIA DE LO EJECUTADO ENTRE LO PLANEADO	

GE004	REPORTE DE CAPTADOS Y REGISTRO DE BENEFICIARIO INGRESADOS AL SISTEMA POR SEMANA	10	FILTRAR POR SEMANA Y VER REPORTE DE INGRESO CAPTADO Y BENEFICIARIO	
-------	---	----	--	--

### 4.3. Plan de Entregas.

Tbl.19: Plan de historias de usuario

ID	NOMBRE	IMPORTANCIA	ESTIMACION
<b>SPRINT 1: INGRESO DEL CAPTADO</b>			
AD001	REGISTRAR DATOS DE CAPTADOS MEDIANTE PAGINA WEB	290	26
<b>SPRINT 2: REGISTRO COMPLETO DEL BENEFICIARIO</b>			
AD002	REGISTRAR DATOS DE BENEFICIARIO ELEGIBLE MEDIANTE PAGINA WEB	280	24
AD003	GENERAR NUMERO DE CONTRATO Y MOSTRAR EN PAG WEB	250	8
<b>SPRINT 3: ACTUALIZACION REGISTROS</b>			
GE004	REPORTE DE CAPTADOS Y REGISTRO DE BENEFICIARIO INGRESADOS AL SISTEMA POR SEMANA	230	10
AD004	ACTUALIZAR DATOS PARA REGISTRO	220	10
AD006	ACTUALIZAR ESTADO PRE FMV	200	8
<b>SPRINT 4: INGRESO DE AHORRO</b>			
AD007	ACTUALIZAR ESTADO POST ENVIO FMV	180	8
AD008	INGRESAR CODIGO FMV AL BENEFICIARIO REGISTRADO	150	10

AD019	INGRESO DE AHORRO POR PAG WEB	140	8
<b>SPRINT 5: BETA CONSTRUCCION</b>			
AS001	ASIGNACION A CARTA FIANZA	120	12
CO001	ACTUALIZAR ESTADO DE OBRA	90	6
<b>SPRINT 6: CUADRO DE EJECUCION</b>			
GE002	CUADRO DE EJECUCION	70	12
<b>SPRINT 7: CUADROS DE ANALISIS</b>			
GE001	CUADRO DE PLANIFICACION	65	15
GE003	CUADRO DE DIFERENCIA	60	10
<b>SPRINT 8: IMPRESIONES DE DOCUMENTOS</b>			
AD005	CARGAR ESCANEOS DE DOCUMENTOS BASE DEL BENEFICIARIO AL SERVIDOR	40	15
CO002	IMPRIMIR DOCUMENTOS BASE DE BENEFICIARIO	30	12

## **4.4. Diseño del sistema**

### **4.4.1. Definición de estándares**

#### **4.4.1.1. Estándar de nomenclaturas para la base de datos**

##### **4.4.1.1.1. Tablas.**

- ✓ Los nombres de las tablas deben ser lo suficientemente descriptivos para identificar los datos que contiene, deben estar en singular e incluir caracteres de la A-Z (mayúsculas). Ejemplo: USUARIO, ZONA, etc.
- ✓ Si está formado por más de una palabra deben ir separadas por un "\_". Ejemplo: ENTIDAD\_FINANCIERA, ENTIDAD\_TECNICA, etc.

##### **4.4.1.1.2. Columnas.**

- ✓ No usar palabras reservadas de transact-sql.
- ✓ Los nombres de las columnas deben ser lo suficientemente descriptivos, pueden ser en singular o plural, e incluir caracteres de la a-z.
- ✓ Los nombres de los campos deben iniciar con la primera letra en mayúsculas y las demás en minúsculas. si existiera otra palabra no se debe separar por ningún carácter especial, debe iniciar con la primera letra en mayúsculas y las demás en

minúsculas. Ejemplo: TelefCelular, CargaFamiliar, etc.

- ✓ Si la columna es clave primaria se usará la siguiente estructura: Id<nombre de la tabla>. Ejemplo: IdArea.

#### **4.4.1.1.3. Restricciones.**

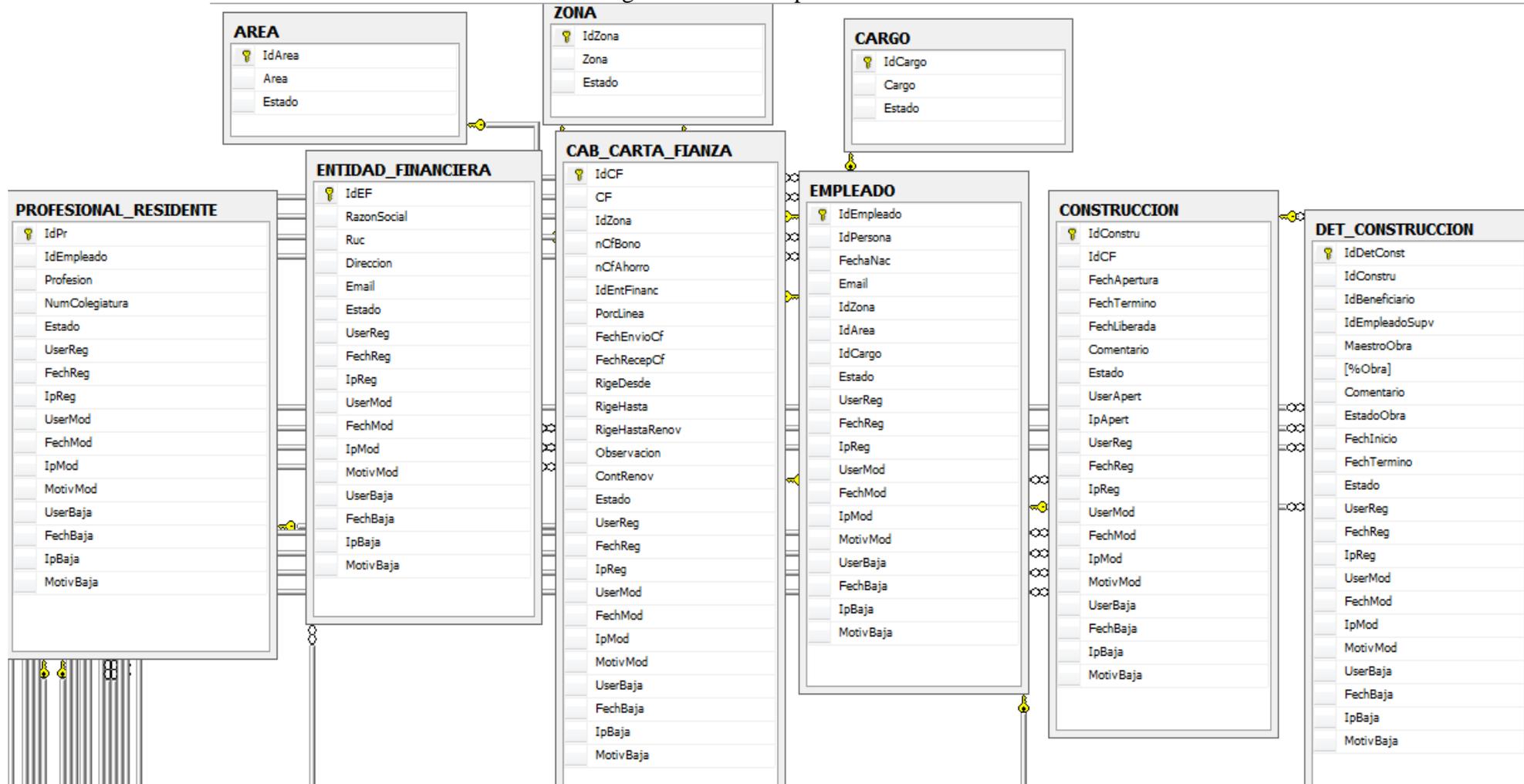
- ✓ Para nombrar las claves primarias se utiliza la siguiente forma:  
<Nombre de tabla que referencia>\_<Campo>\_PK.  
Ejemplo: Area\_IdArea\_PK.
- ✓ Para nombrar las claves foráneas se utiliza la siguiente forma:  
<Nombre de tabla referenciada>\_<Campo>\_FK.  
Ejemplo: Empleado\_IdDistrito\_FK.
- ✓ Al migrar las columnas para formar la clave foránea en la otra tabla pasan con sus mismos nombres.

#### **4.4.1.1.4. Procedimientos almacenados**

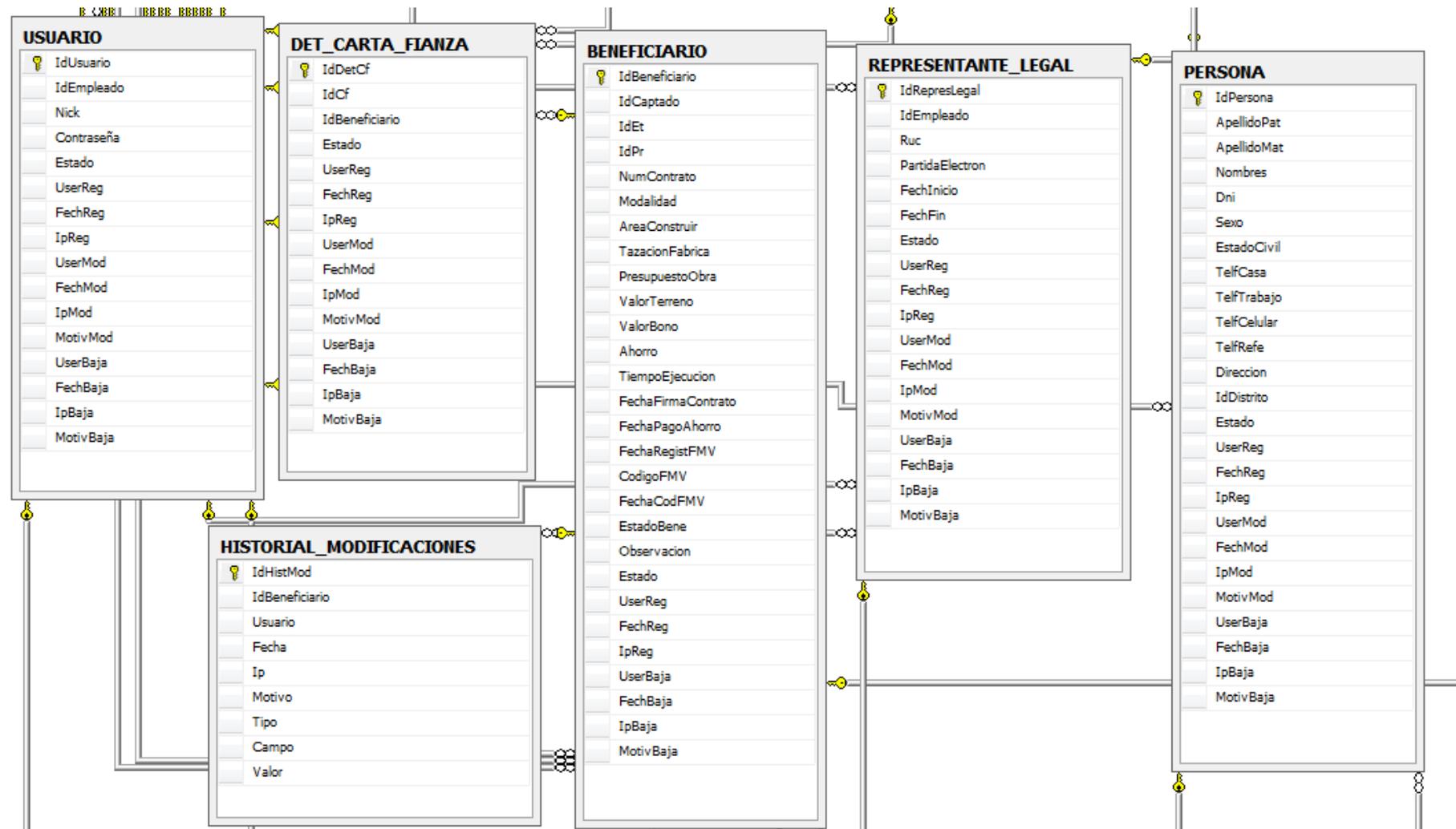
- ✓ Para nombrar los procedimientos almacenados se utiliza la siguiente forma:  
PROC\_<Acción que se va a realizar>\_<Nombre de tabla>.  
Ejemplo: PROC\_INSERTAR\_PERSONA

## 4.5. DISEÑO DE LA BASE DE DATOS

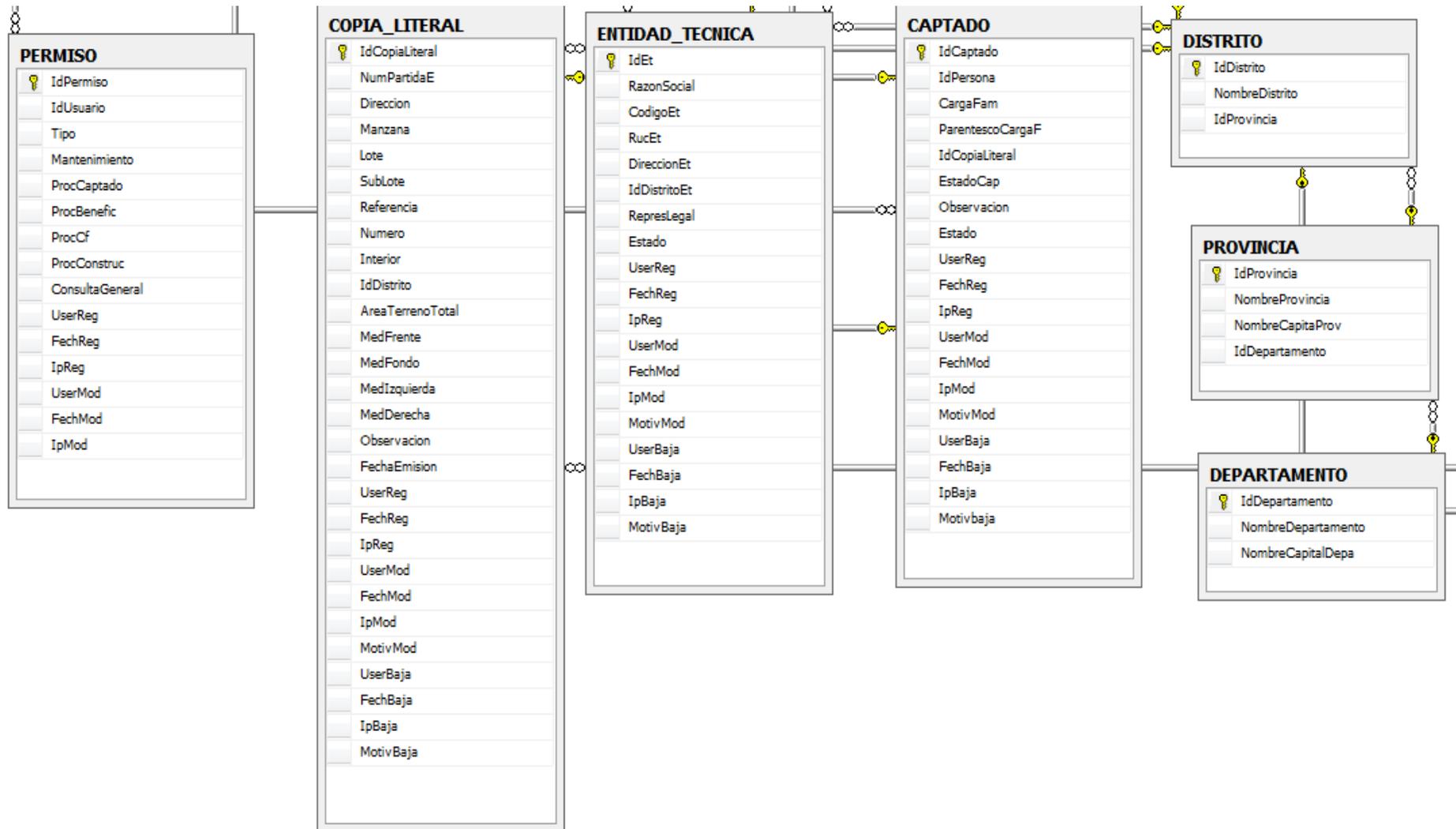
Img.14: Diseño BD part1



Img.15: Diseño BD part2



Img.16: Diseño BD part3



## 4.6. Construcción del sistema por sprint

### 4.6.1. Sprint 1- Ingreso del captado

#### 4.6.1.1. Registrar datos de captado mediante página web.

Img.17: Registrar Captado



### 4.6.2. Sprint 2 – Registro completo de beneficiario

#### 4.6.2.1. Registrar datos beneficiario mediante página web/ número de contrato

Img.18: Registrar beneficiario



### 4.6.3. Sprint 3 – Actualización de registros

#### 4.6.3.1. Reporte de captados y beneficiarios ingresados al sistema por semana

Img.19: Reporte captados

SELECCIONE AÑO: 2014 SELECCIONE SEMANA: 48 [CONSULTAR](#)

NUM	ANIO	SEMANA	DATOS_DEL_CAPTADOR	DATOS_DEL_TITULAR	DNI	ESTADO_CIVIL	SEXO	TELF_C
1	2014	48	INCA GARCIA RUBEN	INGA CADERAS MATEO	98237238	CA	M	092827
2	2014	48	INCA GARCIA RUBEN	AVALOS MORA ASTRID	98769543	DI	F	9865436
3	2014	48	INCA GARCIA RUBEN	CHUMPITAZ CHUMPITAZ FERNANDO	77689463	SO	M	23423
4	2014	48	INCA GARCIA RUBEN	SALAZAR SALAZAR HUDSON WILLIAM	15433029	CA	M	123412
5	2014	48	INCA GARCIA RUBEN	MUNAYA MUNAYA JESSICA ELIZABETH	43250040	CA	F	98765
6	2014	48	INCA GARCIA RUBEN	CHACALIAZA CHACALIAZA JOSE ANDRES	22244736	CA	M	6867
7	2014	48	INCA GARCIA RUBEN	ARTEAGA ARTEAGA ELIZABETH GIULIANA	45794548	CA	F	53634

[DESCARGAR](#)

Img.20: Reporte de beneficiarios

INGRESE DNI DEL BENEFICIARIO:   Todos [Buscar](#)

N	BENEFICIARIO	NUM_CONTRATO	RAZON_SOCIAL	MODALIDAD	AREA_CONSTRUIR	TASACION_FABRICA	PRE
1	ANGELES ALMEYDA NUMERO 2	001-0011-2015	AIF CONTRATISTAS GENERALES SAC	CSP	12.00	12.00	
2	MENESES COLINA MARCO	001-0012-2015	AIF CONTRATISTAS GENERALES SAC	CSP	12.00	12.00	
3	INGA CADERAS MATEO	2014-00001-01	AIF CONTRATISTAS GENERALES SAC	CSP	7.00	8.00	
4	AVALOS MORA ASTRID	2014-00002-01	AIF CONTRATISTAS GENERALES SAC	CSP	32.00	1234.00	
5	CHUMPITAZ CHUMPITAZ FERNANDO	2014-00003-01	AIF CONTRATISTAS GENERALES SAC	CSP	32.00	1234.00	
6	SALAZAR SALAZAR HUDSON WILLIAM	2014-00004-01	AIF CONTRATISTAS GENERALES SAC	CSP	32.00	1234.00	
7	DIAZ DIAZ JIMMY MAXIMO	2014-00005-01	AIF CONTRATISTAS GENERALES SAC	CSP	32.00	1234.00	

1 | 2 [DESCARGAR](#)

### 4.6.3.2. Actualizar datos para registro/Actualizar estado Pre FMV

Img.21: Actualizar beneficiario pre FMV

The screenshot shows a web interface with two main sections. The top section is titled "BENEFICIARIO" and contains a text input field labeled "Ingrese N° de DNI:" followed by a green "Buscar" button. Below this is a large empty rectangular box. The bottom section is titled "ACTUALIZACION DE FECHAS" and contains several input fields. It starts with "Beneficiario:" followed by a text input field, and "Número de Contrato:" followed by another text input field. Below these are two rows of date inputs: "Fecha de Firma del Contrato" and "Fecha de Registro al FMV", each with a colon separator and a date input field. To the right of each date input is a green "Guardar" button.

### 4.6.4. Sprint 4 – Ingreso de ahorro

#### 4.6.4.1. Actualizar estado post envío FMV/Ingresar código FMV al beneficiario

Img.22: Actualizar código FMV

The screenshot shows a web interface with two main sections. The top section is titled "BENEFICIARIO" and contains a text input field labeled "Ingrese N° de Dni:" followed by a green "Buscar" button. Below this is a large empty rectangular box. The bottom section is titled "ACTUALIZACIÓN DE FONDO MI VIVIENDA" and contains three input fields: "Beneficiario:" followed by a text input field, "Código:" followed by a text input field, and "Fecha:" followed by a date input field. Below these fields is a green "Guardar" button.

#### 4.6.4.2. Ingreso de ahorro

Img.23: Ingreso fecha de ahorro

BENEFICIARIO

Ingrese N° de DNI:

---

PAGO DE AHORRO

Beneficiario :

Monto de ahorro :

Fecha de pago de ahorro :

#### 4.6.5. Sprint 5 - Beta construcción

##### 4.6.5.1. Asignación a carta fianza

Img.24: Asignación carta fianza

**CREACIÓN DE CARTA FIANZA**

DATOS DE LA C.F.		BENEFICIARIOS	
CARTA FIANZA	<input type="text"/>	DNI	<input type="text"/> <input type="button" value="+"/>
ZONA	CHINCHA		<input type="button" value="-"/>
N° BONO	<input type="text"/>		
N° AHORRO	<input type="text"/>		
ENT. FINANCIERA	PEPO SAC		
% LINEA	<input type="text"/>		
FEC. ENVÍO	26/02/2015		
FEC. RECEPCIÓN	26/02/2015		
RIGE DESDE	26/02/2015		
RIGE HASTA	26/02/2015		
COMENTARIO	<input type="text"/>		
<input type="button" value="GUARDAR"/> <input type="button" value="LIMPIAR"/> <input type="button" value="CANCELAR"/>			

#### 4.6.5.2. Actualizar estado de obra

Img.25: Actualizar estado de obra

ACTUALIZAR AVANCES DE OBRAS

Carta Fianza:

---

CARTA FIANZA

Carta Fianza :  Fecha Liberación Estimada :   
 Zona :  Porcentaje de Avance :   
 Municipalidades :

---

AVANCE DE OBRAS

Acción	N	BENEFICIARIO	% DE OBRA	SUPERVISOR	MAESTRO DE OBRA	ESTADO DE OBRA	COMENTARIO	FECHA DE INICIO	FECHA DE TERMINO	FECHA TERMINO ESTIMADO
<a href="#">Seleccionar</a>	1	DIAZ DIAZ JIMMY MAXIMO								
<a href="#">Seleccionar</a>	2	SALAZAR SALAZAR HUDSON WILLIAM								
<a href="#">Seleccionar</a>	3	YANQUI YANQUI GLADYS ERLINDA	13	MATEO INGA	FELIPITO	ANTES DE TECHO		11/02/2015 12:00:00 a.m.		04/03/2015 12:00:00 a.m.

Actualizar obra de: [YANQUI YANQUI GLADYS ERLINDA](#)

**Encargados :**

Supervisor :  Maestro de obra :

**Fechas :**

Inicio de obra :  Termino estimado :

#### 4.6.6. Sprint 6 – Cuadro de ejecución

##### 4.6.6.1. Cuadro de ejecución

Img.26: Cuadro de ejecución

SELECCIONE AÑO:  SELECCIONE MES:

ZONA	TOTAL	POR_INICIAR	CONSTRUYENDO	TERMINADOS
CHINCHA	6	2	1	3
PISCO	1	1	0	0

#### 4.6.7. Sprint 7 – Cuadro de análisis

##### 4.6.7.1. Cuadro de planificación / Cuadro de diferencia

Img.27: Cuadro de planificación/ Cuadro de diferencia



SELECCIONE AÑO: Todos SELECCIONE MES: Consultar

ZONA	PLANIFICADOS	TERMINADOS	DIFERENCIA
CHINCHA	6	3	3
PISCO	1	0	1

DESCARGAR

##### 4.6.7.2. Sprint 8 – Impresiones de documentos

No se realizó este sprint por el motivo de falta de tiempo, pero la falta de este último sprint no afecta a los procesos principales del sistema ya que dentro del plan de entrega se previó la posible falta del sprint por ese motivo las historias de usuarios están de color verde.

## 5. CONCLUSIONES

- ✓ Los Sistemas Web facilitan y optimizan los procesos en las organizaciones brindándoles una mejor toma de decisiones en sus procesos, con el fin de dar un mejor servicio a los clientes como a sus trabajadores.
  
- ✓ El diseño y aplicación de un Programa de Gestión de información en la empresa, mejora sensiblemente el control y organización de la información, y contribuye al logro de los objetivos empresariales.
  
- ✓ El éxito de la implementación de un Sistema Web, involucra varios aspectos en los cuales la capacitación previa del personal que va utilizar el programa es un punto crítico para cumplir con los objetivos.
  
- ✓ La evaluación de los recursos tanto técnicos como humanos es muy importante, ya que, si no se cuenta con ellos no se puede desarrollar el Sistema Web.
  
- ✓ La metodología Scrum nos permite obtener un software de calidad y garantía, ya que ayudan a gestionar los cambios que se puedan presentar durante el desarrollo del sistema.
  
- ✓ La participación del cliente ha sido un aspecto clave en el éxito del planeamiento y desarrollo del sistema. Las reuniones que se realizaron ayudaron a la definición y validación de los requerimientos del sistema.

## **6. RECOMENDACIONES**

- ✓ Es recomendable que todas las empresas diseñen y apliquen un programa de gestión de información que les permita controlar y organizar toda la información que adquieren durante distintos procesos, ya que ello contribuye al logro de sus objetivos empresariales.
  
- ✓ Es importante que todas las áreas internas de la institución, involucradas en la generación y producción de información, realicen coordinaciones para uniformizar los procedimientos de trámite documentario y evitar la pérdida de tiempo.
  
- ✓ Se recomienda a la organización brindar una capacitación en temas de T.I. para que el personal tenga un mayor conocimiento en su uso y acelerar los beneficios que brinda en su implementación.
  
- ✓ Se debe tener en cuenta un plan o programa de mantenimiento tanto para el equipo de cómputo como para el software. Ya que, los requerimientos como las necesidades de la institución van cambiando, el software, también, debe hacerlo. Y si un equipo no se encuentra en óptimas condiciones repercutirá en el desempeño del software.
  
- ✓ Realizar una evaluación periódica de la interacción entre el usuario y el sistema, con el objetivo de realizar mejoras en las funcionalidades del sistema.

## 7. REFERENCIAS BIBLIOGRÁFICAS

## 7.1. Bibliografía

- ✓ Francisco Otero, Microsoft Visual Studio 2010 desde cero. España: CampusMVP: 2005.
- ✓ Fernando Giardina, Guía de ASP.NET- Desarrollo de sitios y aplicaciones web dinámicas. Argentina: Maestros Web
- ✓ Juan Quijano, Configurando un IIS para .NET. España: Genbetadev : 2011
- ✓ Silberschats/Korth. Fundamentos de Base de Datos. España: Mc Graw Hill
- ✓ Henrik Kniberg, Scrum y Xp desde las trincheras. Suecia: InfoQ : 2007
- ✓ J. Ferguson / B. Patterson. La Biblia de C#. España : 2002
- ✓ Laurent Debrauwer. Patrones de diseño para C#, Los 23 modelos de diseño.
- ✓ Manuel Alarcón. Programación Web con Visual Studio y asp.net. España: Microsoft MVP.
- ✓ Kent/Beck. Una explicación de la programación extrema: aceptar el cambio. España: ADDISON-WESLEY: 2002.
- ✓ Francisco Charte, Sql Server 2012. : Anaya Multimedia: 2012.

## 7.2. Linkografía

- ✓ Henrik Kniberg (2007). Scrum y Xp desde las trincheras <http://www.buenastareas.com/ensayos/Tesis-Sistema-Web-Para-La-Gesti%C3%B3n/1419488.html>
  
- ✓ Jorge Ercoli (2007). Arquitectura de sistemas informáticos <http://metodologiasdesistemas.blogspot.com/2007/05/diseo-en-3-capas-fisicas-lgicas-es.html>
  
- ✓ Wikipedia (2014). AJAX. <http://es.wikipedia.org/wiki/AJAX>
  
- ✓ Asp.net (2014). Ajax control kit <http://www.ajaxcontroltoolkit.com/>
  
- ✓ Kenneth Hamlett, Demand Media (2014). Las características de un sistema de gestión de información <http://pyme.lavoztx.com/las-caractersticas-de-un-sistema-de-gestin-de-informacin-9451.html>
  
- ✓ Wikipedia (2014). Microsoft SQL Server [http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server)

## 8. ANEXOS

## 8.1. Anexo N° 1

MATRIZ DE CONSISTENCIA			
PROBLEMA	OBJETIVOS	HIPOTESIS	METODOLOGIA
<p><b>Problema general:</b></p> <p>¿Se logrará mediante la sistematización de los procesos una gestión e integridad de información, entre las áreas de administración y construcción de la empresa constructora "AIF CONSTRATISTAS GENERALES S.A.C"?</p>	<p><b>Objetivo general:</b></p> <p>La gestión de la información para lograr salvaguardar la integridad y reutilización de los datos dentro de procesos de trámites documentarios y auditorios de la empresa constructora "AIF CONSTRATISTAS GENERALES S.A.C".</p>	<p><b>Hipótesis general:</b></p> <p>Mediante la sistematización de procesos se lograra la gestión de la información que influye significativamente para la correcta integridad en los trámites documentarios y control entre las áreas de administración y construcción de la empresa constructora "AIF CONSTRATISTAS GENERALES S.A.C".</p>	<p><b>Metodología general:</b></p> <p>El proyecto de investigación es de tipo aplicada porque su nivel es experimental, y tiene por objetivo la comprobación de hipótesis, donde se manipula o controla la variable independiente (Sistema Informático) y Comprobar su influencia en la variable dependiente (Centralización de información).</p>

## 8.2. Anexo N° 2

# MANUAL DE USUARIO

## **MANUAL DE USUARIO**

### **1. INTRODUCCIÓN**

En este documento se describirá los objetivos e información clara y concisa de cómo utilizar el **Sistema de Gestión de información SCAIF**, así mismo el **funcionamiento** para la web y escritorio.

El **Sistema de Gestión de información SCAIF** fue creado para brindar facilidades en el manejo de la gestión de la información para la entidad técnica **AIF CONTRATISTAS GENERALES S.A.C.**

El personal de la empresa podrá registrar los datos de los beneficiarios, consultar el estado de los beneficiarios, exportar e imprimir reportes necesarios para controlar el avance en los procesos por beneficiario y en general. Así mismo los empleados de las sedes podrán realizar un seguimiento de sus beneficiarios sin necesidad de acercarse a la oficina central.

Es de mucha importancia consultar este manual antes y/o durante la visualización de las páginas, ya que lo guiará paso a paso en el manejo de las funciones en él.

Con el fin de facilitar la comprensión del manual, se incluye gráficos explicativos.

### **2. LO QUE DEBE CONOCER**

Los conocimientos mínimos que deben tener las personas que operarán las páginas y deberán utilizar este manual son:

- Conocimientos básicos de Navegación en Web.
- Conocimiento básico de Internet.
- Conocimiento básico de Windows.

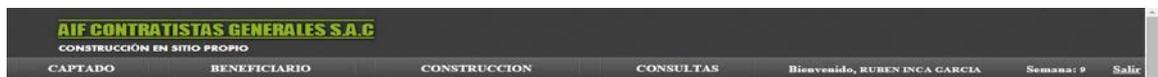
### 3. SISTEMA WEB DE GESTIÓN DE LA INFORMACION SCAIF

#### a. INGRESO AL SISTEMA DE GESTIÓN DE GESTION DE LA INFORMACION SCAIF

- ✓ Ingrese el nombre Usuario y la Contraseña.



- Tenemos el ingreso a la página principal donde podremos ingresar al menú general.



#### b. PROCESOS DE LAS ÁREAS IMPLICADAS

**“Ingresar Persona”:** Sobre “CAPTADO” acceder a la pestaña “PERSONA”-Ingresar Persona:  
Si haces clic en “Ingresar Persona” se abrirá el formato donde deberás llenar los datos básicos para ingresar a la persona donde algunos datos son obligatorios de requerirlos se solicitaran.

DATOS DE LA PERSONA	
Apellido Paterno	<input type="text"/>
Apellido Materno	<input type="text"/>
Nombre(s)	<input type="text"/>
DNI	<input type="text"/>
Sexo	<input type="radio"/> Masculino <input type="radio"/> Femenino
Estado Civil	<input type="text" value="CASADO(A)"/>
DATOS REFERENCIALES	
N° Teléfono de Casa	<input type="text"/>
N° Teléfono del Centro de Trabajo	<input type="text"/>
N° Teléfono Celular	<input type="text"/>
N° Teléfono de Referencia	<input type="text"/>
DATOS DE DOMICILIO	
Departamento	Seleccione Depart... <input type="text"/>
Provincia	<input type="text"/>
Distrito	<input type="text"/>
<input type="button" value="GUARDAR"/> <input type="button" value="BORRAR"/>	

**“Modificar Persona”:** Sobre “CAPTADO” acceder a la pestaña “PERSONA”-Modificar Persona:  
Si haces clic en “Modificar Persona” se abrirá el formato donde deberás llenar los datos que deseas modificar de la persona.

**PERSONA**

Ingrese N° de DNI de la persona:

Apellido Paterno : <input type="text"/>	DNI : <input type="text"/>
Apellido Materno : <input type="text"/>	N° Teléfono de Casa : <input type="text"/>
Nombre(s) : <input type="text"/>	N° Teléfono del Centro de Trabajo : <input type="text"/>
Sexo : <input checked="" type="radio"/> Masculino <input type="radio"/> Femenino	N° Teléfono Celular : <input type="text"/>
Estado Civil : CASADO(A) ▼	N° Teléfono de Referencia : <input type="text"/>

---

**DATOS DE DOMICILIO**

Ubigeo Actual : <input type="text"/>	Provincia : <input type="text"/>	Distrito : <input type="text"/>
Departamento : <input type="text"/>		
Dirección : <input type="text"/>		

**“Ingresar Captado”**: Si haces clic en “Ingresar Captado” se abrirá el formato donde deberás llenar los datos básicos para registrar a tu Captado donde algunos datos son obligatorios de requerirlos se solicitaran.

<p style="text-align: center;"><b>PERSONA</b></p> <p style="text-align: center;">Ingrese N° de DNI de la persona : <input type="text"/> <input type="button" value="Buscar"/></p> <table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Apellido Paterno : <input type="text"/></td> <td style="width: 50%;">N° Teléfono de Casa : <input type="text"/></td> </tr> <tr> <td>Apellido Materno : <input type="text"/></td> <td>N° Teléfono del Centro de Trabajo : <input type="text"/></td> </tr> <tr> <td>Nombre(s) : <input type="text"/></td> <td>N° Teléfono Celular : <input type="text"/></td> </tr> <tr> <td>Sexo : <input checked="" type="radio"/> Masculino <input type="radio"/> Femenino</td> <td>N° Teléfono de Referencia : <input type="text"/></td> </tr> <tr> <td>Estado Civil : <input type="text"/></td> <td></td> </tr> </table> <hr/> <p style="text-align: center;"><b>CARGA FASELIDE</b></p> <p style="text-align: center;">Ingrese N° de DNI de la persona : <input type="text"/> <input type="button" value="Buscar"/></p> <table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Apellido Paterno : <input type="text"/></td> <td style="width: 50%;">N° Teléfono de Casa : <input type="text"/></td> </tr> <tr> <td>Apellido Materno : <input type="text"/></td> <td>N° Teléfono del Centro de Trabajo : <input type="text"/></td> </tr> <tr> <td>Nombre(s) : <input type="text"/></td> <td>N° Teléfono Celular : <input type="text"/></td> </tr> <tr> <td>Sexo : <input type="text"/></td> <td>N° Teléfono de Referencia : <input type="text"/></td> </tr> <tr> <td>Estado Civil : <input type="text"/></td> <td>Personas : SELECCIONE *</td> </tr> </table> <hr/> <p style="text-align: center;"><b>COPIA LITERAL</b></p> <p style="text-align: center;">N° de Permiso : <input type="text"/> Fecha de Emisión : 28/02/2015 Fecha de Vencimiento : 28/02/2015</p> <p>Departamento : Seleccione Departamento ▼ Provincia : <input type="text"/> Distrito : <input type="text"/></p> <p>Dirección : <input type="text"/> Referencia : <input type="text"/></p> <p>Materia : <input type="text"/> Lote : <input type="text"/> Sub Lote : <input type="text"/> Sitios : <input type="text"/> Interiores : <input type="text"/></p> <p><b>Medidas del Terreno:</b></p> <p>Area Total : <input type="text"/> Med. Frente : <input type="text"/> Med. Derecha : <input type="text"/> Med. Izquierda : <input type="text"/> Med. Fondo : <input type="text"/></p> <hr/> <p style="text-align: center;"><b>OBSERVACIONES</b></p> <p>Estado de Captado : ELEGIBLE ▼ Observación de Captado : <input type="text"/></p> <p>Estado de Copia Literal : VIGENTE Observación de Copia Literal : <input type="text"/></p> <p style="text-align: right;"><input type="button" value="Agregar Captado"/></p>	Apellido Paterno : <input type="text"/>	N° Teléfono de Casa : <input type="text"/>	Apellido Materno : <input type="text"/>	N° Teléfono del Centro de Trabajo : <input type="text"/>	Nombre(s) : <input type="text"/>	N° Teléfono Celular : <input type="text"/>	Sexo : <input checked="" type="radio"/> Masculino <input type="radio"/> Femenino	N° Teléfono de Referencia : <input type="text"/>	Estado Civil : <input type="text"/>		Apellido Paterno : <input type="text"/>	N° Teléfono de Casa : <input type="text"/>	Apellido Materno : <input type="text"/>	N° Teléfono del Centro de Trabajo : <input type="text"/>	Nombre(s) : <input type="text"/>	N° Teléfono Celular : <input type="text"/>	Sexo : <input type="text"/>	N° Teléfono de Referencia : <input type="text"/>	Estado Civil : <input type="text"/>	Personas : SELECCIONE *	<p style="text-align: center;"><b>PERSONA</b></p> <p style="text-align: center;">Ingrese N° de DNI de la persona : <input type="text"/> <input type="button" value="Buscar"/></p> <table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Apellido Paterno : <input type="text"/></td> <td style="width: 50%;">DNI : <input type="text"/></td> </tr> <tr> <td>Apellido Materno : <input type="text"/></td> <td>N° Teléfono de Casa : <input type="text"/></td> </tr> <tr> <td>Nombre(s) : <input type="text"/></td> <td>N° Teléfono del Centro de Trabajo : <input type="text"/></td> </tr> <tr> <td>Sexo : <input checked="" type="radio"/> Masculino <input type="radio"/> Femenino</td> <td>N° Teléfono Celular : <input type="text"/></td> </tr> <tr> <td>Estado Civil : CASADO(A) ▼</td> <td>N° Teléfono de Referencia : <input type="text"/></td> </tr> </table> <hr/> <p style="text-align: center;"><b>DATOS DE DOMICILIO</b></p> <table border="0" style="width: 100%;"> <tr> <td>Ubigeo Actual : <input type="text"/></td> <td>Provincia : <input type="text"/></td> <td>Distrito : <input type="text"/></td> </tr> <tr> <td>Departamento : <input type="text"/></td> <td></td> <td></td> </tr> <tr> <td>Dirección : <input type="text"/></td> <td></td> <td></td> </tr> </table> <p style="text-align: center;"><input type="button" value="Actualizar"/></p>	Apellido Paterno : <input type="text"/>	DNI : <input type="text"/>	Apellido Materno : <input type="text"/>	N° Teléfono de Casa : <input type="text"/>	Nombre(s) : <input type="text"/>	N° Teléfono del Centro de Trabajo : <input type="text"/>	Sexo : <input checked="" type="radio"/> Masculino <input type="radio"/> Femenino	N° Teléfono Celular : <input type="text"/>	Estado Civil : CASADO(A) ▼	N° Teléfono de Referencia : <input type="text"/>	Ubigeo Actual : <input type="text"/>	Provincia : <input type="text"/>	Distrito : <input type="text"/>	Departamento : <input type="text"/>			Dirección : <input type="text"/>		
Apellido Paterno : <input type="text"/>	N° Teléfono de Casa : <input type="text"/>																																							
Apellido Materno : <input type="text"/>	N° Teléfono del Centro de Trabajo : <input type="text"/>																																							
Nombre(s) : <input type="text"/>	N° Teléfono Celular : <input type="text"/>																																							
Sexo : <input checked="" type="radio"/> Masculino <input type="radio"/> Femenino	N° Teléfono de Referencia : <input type="text"/>																																							
Estado Civil : <input type="text"/>																																								
Apellido Paterno : <input type="text"/>	N° Teléfono de Casa : <input type="text"/>																																							
Apellido Materno : <input type="text"/>	N° Teléfono del Centro de Trabajo : <input type="text"/>																																							
Nombre(s) : <input type="text"/>	N° Teléfono Celular : <input type="text"/>																																							
Sexo : <input type="text"/>	N° Teléfono de Referencia : <input type="text"/>																																							
Estado Civil : <input type="text"/>	Personas : SELECCIONE *																																							
Apellido Paterno : <input type="text"/>	DNI : <input type="text"/>																																							
Apellido Materno : <input type="text"/>	N° Teléfono de Casa : <input type="text"/>																																							
Nombre(s) : <input type="text"/>	N° Teléfono del Centro de Trabajo : <input type="text"/>																																							
Sexo : <input checked="" type="radio"/> Masculino <input type="radio"/> Femenino	N° Teléfono Celular : <input type="text"/>																																							
Estado Civil : CASADO(A) ▼	N° Teléfono de Referencia : <input type="text"/>																																							
Ubigeo Actual : <input type="text"/>	Provincia : <input type="text"/>	Distrito : <input type="text"/>																																						
Departamento : <input type="text"/>																																								
Dirección : <input type="text"/>																																								

**“Modificar Captado”**: Si haces clic en “Modificar Captado” se abrirá el formato donde llenaras los datos para modificar a tu Captado o su copia literal.

TITULAR

Ingreso N° de DNI de la persona :  Buscar

Datos de Captado :

---

CARGA FAMILIAR

Ingreso N° de DNI de la persona :  Cambiar

Apellido Paterno :  N° Títulos de Casa :

Apellido Materno :  N° Títulos de Cursos de Trabajo :

Nombres :  N° Títulos de Cédula :

Sexo :  N° Títulos de Referencia :

Estado Civil :  Permisos : SELECCIONAR

---

N° de Parcela :  Fecha de Emisión :  Fecha de Emisión :

Ciudad :

Departamento :  Provincia :  Distrito :

Division :  Referencia :

Manzana :  Lote :  Sub Lote :  Número :  Sector :

Medidas del Terreno:

Área Total :  Mtd. Frente :  Mtd. Fondo :  Mtd. Ancho :  Mtd. Largo :  Mtd. Fondo :

---

OBSERVACIONES

Estado de Captado : ELEGIBLE Observación de Captado :

Estado de Copia Literal :  Observación de Copia Literal :

Guardar Cambios Cop. Literal

“**CAPTADO REPORTE**”. Si ingresas aquí se podrá visualizar y exportar un reporte de los captados del usuario que ingreso si desea por año y semana.

SELECCIONA AÑO: 2014 SELECCIONA SEMANA: Todas CONSULTAR

NUM	ANIO	SEMANA	DATOS_DEL_CAPTADOR	DATOS_DEL_TITULAR	DNI	ESTADO_CIVIL	SEXO	TELF_C
1	2014	48	INCA GARCIA RUBEN	INGA CADERAS MATEO	98237238	CA	M	092827
2	2014	48	INCA GARCIA RUBEN	AVALOS MORA ASTRID	98769543	DI	F	9865438
3	2014	48	INCA GARCIA RUBEN	CHUMPITAZ CHUMPITAZ FERNANDO	77689463	SO	M	23423
4	2014	48	INCA GARCIA RUBEN	SALAZAR SALAZAR HUDSON WILLIAM	15433029	CA	M	123412
5	2014	48	INCA GARCIA RUBEN	MUNAYA MUNAYA JESSICA ELIZABETH	43250040	CA	F	98765
6	2014	48	INCA GARCIA RUBEN	CHACALIAZA CHACALIAZA JOSE ANDRES	22244736	CA	M	6867
7	2014	48	INCA GARCIA RUBEN	ARTEAGA ARTEAGA ELIZABETH GIULIANA	45794548	CA	F	53634

1 | 2

DESCARGAR

“**DESCARGAR DIGITAL**”. Si ingresas aquí se podrá visualizar y descargar reporte de los captados para entrega a oficina si desea por año y semana.

SELECCIONA AÑO: 2014 SELECCIONA SEMANA: 48 CONSULTAR

DATOS_DEL_TITULAR	DNI	EstadoCap	DEPARTAMENTO	PROVINCIA	DISTRITO	Dirección	ESTAD
INGA CADERAS MATEO	98237238	BENEFICIARIO	ICA	ICA	PUEBLO NUEVO	SAN JUAN	
AVALOS MORA ASTRID	98769543	ELEGIBLE	ICA	CHINCHA	CHINCHA ALTA	PRIMERO DE JUNIO	
CHUMPIFAZ CHUMPIFAZ FERNANDO	77689463	ELEGIBLE	ICA	PISCO	PISCO	AA-PH HABILITACION URBANA ALTO EL MOLINO SECTOR VII	
SALAZAR SALAZAR HUDSON WILLIAM	15433029	ELEGIBLE	ICA	PISCO	PISCO	AA-PH HABILITACION URBANA ALTO EL MOLINO SECTOR VII	
MUNAYA MUNAYA JESSICA ELIZABETH	43250040	ELEGIBLE	ICA	PISCO	PISCO	AA-PH HABILITACION URBANA ALTO EL MOLINO SECTOR VII	
CHACALAZA CHACALAZA JOSE ANDRES	22244736	ELEGIBLE	ICA	PISCO	PISCO	AA-PH HABILITACION URBANA ALTO EL MOLINO SECTOR VII	
ARTEAGA ARTEAGA ELIZABETH GIULIANA	45794548	ELEGIBLE	ICA	PISCO	PISCO	AA-PH HABILITACION URBANA ALTO EL MOLINO SECTOR VII	

DESCARGAR

“Ingresar Beneficiario”. Si haces clic en “BENEFICIARIO” “Ingresar Beneficiario” se abrirá el formato donde deberás llenar los datos básicos para pasar de un CAPTADO A UN BENEFICIARIO donde algunos datos son obligatorios de requerirlos se solicitaran.

CAPTADO

Ingrese N° de DNI:  Buscar

Acción	IdCaptado	ApellidoPat	ApellidoMat	Nombres	Dni	EstadoCivil	CargaFamiliar	EstadoCap	Ob
Seleccionar	CAP0001	INGA	CADERAS	MATEO	98237238	CA	NIESTRO	BENEFICIARIO	

BENEFICIARIO

Beneficiario:  Profesional Residente: INCA GARCIA RUBEN

Entidad Técnica: AIF CONTRATISTAS GEN \* Modalidad: CSP

Área Construir:  Observación:

Taxación de Fabrica:

Presupuesto:

Valor del Terreno:

Valor del Bono:

Ahorro:

Tiempo de Ejecución:

Generar N° Contrato y Guardar

“Modificar Beneficiario”. Si haces clic en “BENEFICIARIO” “Beneficiario Modificar” se abrirá el formato donde deberás llenar los datos para modificar.

BENEFICIARIO

Ingrese N° de DNI: 98237238 Buscar

Acción	IdBeneficiario	ApellidoPat	ApellidoMat	Nombres	RazonSocial	Profesional Rest	Modalidad	Ape
Seleccionar	BEN0001	INGA	CADERAS	MATEO	AIF CONTRATISTAS GENERALES SAC	INCA GARCIA RUBEN	CSP	

DATOS

Beneficiario:  Profesional Residente: INCA GARCIA RUBEN

Entidad Técnica: AIF CONTRATISTAS GEN \* Modalidad: CSP

Área Construir:  Observación:

Taxación de Fabrica:

Presupuesto:

Valor del Terreno:

Valor del Bono:

Ahorro:

Tiempo de Ejecución:

Guardar Cambios

“Actualización de fechas”. Si haces clic en “BENEFICIARIO” “Ingresar Beneficiario” se abrirá el formato donde deberás llenar los datos básicos para registrar las fechas de firmas donde algunos datos son obligatorios de requerirlos se solicitaran.

BENEFICIARIO

Ingrese N° de DNI:

	IdBeneficiario	ApellidoPat	ApellidoMat	Nombres	Dni	zona	NumContrato	Observacion	Es
<a href="#">Seleccionar</a>	BEN0001	INGA	CADERAS	MATEO	98237238	CHINCHA	2014-00001-01		

ACTUALIZACION DE FECHAS

Beneficiario:  Número de Contrato:

Fecha de Firma del Contrato:

Fecha de Registro al FMV:

**“Ingresar Código”**. Si haces clic en “Ingresar Código” se abrirá el formato donde deberás llenar los datos básicos para REGISTRAR el CÓDIGO de tu BENEFICIARIO donde algunos datos son obligatorios de requerirlos se solicitaran

ACTUALIZACION DE FONDO MI VIVIENDA

Ingrese N° de Dni:

	IdBeneficiario	ApellidoPat	ApellidoMat	Nombres	Dni	NumContrato	Observacion	EstadoBene
<a href="#">Seleccionar</a>	BEN0001	INGA	CADERAS	MATEO	98237238	2014-00001-01		CODIGO

Beneficiario:

Código:

Fecha:

**“Apertura de Carta Fianza”**. Una vez este creada la carta fianza (CONSULTE MANUAL DE ESCRITORIO) y este apta por vía web se apertura en el área de construcción.

Carta Fianza:

---

DETALLE DE CARTA FIANZA

Municipalidad:

[Beneficiario:](#)

**“Avance de obra”**. En este módulo se actualizara diariamente el avance de las obras por carta fianza detallada por beneficiario.

-ACTUALIZAR AVANCES DE OBRAS-

Carta Fianza:

-CARTA FIANZA-

Carta Fianza :	<input type="text"/>	Fecha Liberación Estimada :	<input type="text"/>
Zona :	<input type="text"/>	Porcentaje de Avance :	<input type="text"/>
Municipalidades :	<input type="text"/>		

-AVANCE DE OBRAS-

**“Consultar información de la persona”**. Aquí se podrá visualizar por medio de una consulta el estado detallado en que se encuentre la persona beneficiaria.

Ingrese Dni de la persona a consultar:

INFORMACIÓN DE PERSONA	
<b>DATOS DE LA PERSONA</b>	
Fecha de Registro: 17/11/2014 12:22:11 p.m.	
APELLIDOS Y NOMBRES:	INGA CADERAS MATEO
DNI:	98237238
SEXO:	M
ESTADO CIVIL:	CA
N° DE CONTACTO:	09282782 - 82383292 - 98237237 - 28738782
DIRECCIÓN:	LOS ROSALES
UBIGEO:	ICA - CHINCHA - CHINCHA ALTA
INFORMACIÓN DE CAPTADO	
INFORMACIÓN DE BENEFICIARIO	
INFORMACIÓN DE CONSTRUCCIÓN	

**“Cuadro de Ejecución”**. Si se accede aquí se podrá consultar el cuadro general de ejecución o por filtros de año y mes.

SELECCIONE AÑO:  SELECCIONE MES:

ZONA	TOTAL	POR_INICIAR	CONSTRUYENDO	TERMINADOS
CHINCHA	6	2	1	3
PISCO	1	1	0	0

**“Cuadro de planificación y diferencia”**. Si se accede aquí se puede consultar las construcciones planificadas vs los terminados y mostrándonos finalmente la diferencia.

SELECCIONE AÑO:  SELECCIONE MES:

ZONA	PLANIFICADOS	TERMINADOS	DIFERENCIA
CHINCHA	6	3	3
PISCO	1	0	1

#### 4. SISTEMA DE ESCRITORIO

**Paso: número 1:** Ingresar al sistema con usuario y contraseña.



**Paso: número 2:** Si haces clic en “PERSONA - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para ingresar a la persona donde algunos datos son obligatorios de requerirlos se solicitaran.



- Si haces clic en “EMPLEADO - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para registrar a un empleado donde algunos datos son obligatorios de requerirlos se solicitaran.

REGISTRAR EMPLEADO

PERSONA

DNI

NOMBRE(S)

DATOS DE TRABAJO

ZONA: CHINCHA

AREA: VENTAS

CARGO: CAPTADOR

DATOS ADICIONALES

FECH. DE NACIMIENTO: 09/01/1997

EMAIL

GUARDAR LIMPIAR CANCELAR

- Si haces clic en “PROFESIONAL RESIDENTE - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para al profesional residente donde algunos datos son obligatorios de requerirlos se solicitaran.

REGISTRAR PROFESIONAL RESIDENTE

EMPLEADO

DNI

NOMBRE(S)

DATOS DE PROFESION

PROFESION

REGISTRO

GUARDAR LIMPIAR CANCELAR

- Si haces clic en “REPRESENTANTE LEGAL - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para registrar a un representante legal donde algunos datos son obligatorios de requerirlos se solicitaran.

REGISTRAR REPRESENTANTE LEGAL

EMPLEADO

DNI

NOMBRE(S)

DATOS DEL REPRESENTANTE

RUC

P. ELECTRONICA

GUARDAR LIMPIAR CANCELAR

- Si haces clic en “ENTIDAD FINANCIERA - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para registrar a

una entidad financiera donde algunos datos son obligatorios de requerirlos se solicitaran.

- Si haces clic en “ENTIDAD TÉCNICA - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para registrar su entidad técnica donde algunos datos son obligatorios de requerirlos se solicitaran.

- Si haces clic en “ÁREA - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para registrar un área donde algunos datos son obligatorios de requerirlos se solicitaran.

- Si haces clic en “CARGO - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para registrar a un cargo donde algunos datos son obligatorios de requerirlos se solicitaran.



- Si haces clic en “ZONA - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para registrar la zona donde algunos datos son obligatorios de requerirlos se solicitaran.



- Si haces clic en “USUARIO - REGISTRAR” se abrirá el formato donde deberás llenar los datos básicos para registrar a un usuario donde algunos datos son obligatorios de requerirlos se solicitaran.



- SOLO PARA EL ADMINISTRADOR: Aquí es donde por usuarios podrá darles permisos a los accesos y permisos al sistema.



**Al escoger el menú de CARTA FIANZA (CREAR CARTA FIANZA – RENOVAR CARTA FIANZA - CONSULTAS)**

- Si haces clic en “**CREAR CARTA FIANZA**” se abrirá el formato donde deberás llenar los datos básicos para registrar a un empleado donde algunos datos son obligatorios de requerirlos se solicitaran.



- Si haces clic en “**RENOVAR CARTA FIANZA**” se abrirá el formato donde deberás llenar los datos básicos para poder renovar una carta fianza vencida donde algunos datos son obligatorios de requerirlos se solicitaran .



### **8.3. ANEXO N° 3**

## **MANUAL DEL PROGRAMADOR**

## MANUAL DEL PROGRAMADOR

### I. Estructura de Base de Datos

Nombre de la Base de Datos: SCAIF\_PERM

```
CREATE DATABASE SCAIF_PERM
```

```
USE [SCAIF_PERM]
```

```
--ESTRUCTURA DE LA TABLA : [USUARIO]
```

```
CREATE TABLE [dbo].[USUARIO](  
[IdUsuario] [char](5) NOT NULL,  
[IdEmpleado] [char](6) NOT NULL,  
[Nick] [nvarchar](12) NOT NULL,  
[Contraseña] [nvarchar](12) NOT NULL,  
[Estado] [char](1) NOT NULL,  
[UserReg] [char](5) NOT NULL,  
[FechReg] [datetime] NOT NULL,  
[IpReg] [varchar](15) NOT NULL,  
[UserMod] [char](5) NULL,  
[FechMod] [datetime] NULL,  
[IpMod] [varchar](15) NULL,  
[MotivMod] [nvarchar](60) NULL,  
[UserBaja] [char](5) NULL,  
[FechBaja] [datetime] NULL,  
[IpBaja] [varchar](15) NULL,  
[MotivBaja] [nvarchar](60) NULL
```

```
--ESTRUCTURA DE LA TABLA : [PERMISO]
```

```
CREATE TABLE [dbo].[PERMISO](  
[IdPermiso] [int] IDENTITY(1,1) NOT NULL,  
[IdUsuario] [char](5) NOT NULL,  
[Tipo] [nvarchar](20) NOT NULL,  
[Mantenimiento] [char](1) NULL,  
[ProcCaptado] [char](1) NULL,  
[ProcBenefic] [char](1) NULL,  
[ProcCf] [char](1) NULL,  
[ProcConstruc] [char](1) NULL,  
[ConsultaGeneral] [char](1) NULL,  
[UserReg] [char](5) NOT NULL,  
[FechReg] [datetime] NOT NULL,  
[IpReg] [varchar](15) NOT NULL,  
[UserMod] [char](5) NULL,  
[FechMod] [datetime] NULL,  
[IpMod] [varchar](15) NULL
```

```
--ESTRUCTURA DE LA TABLA : [ENTIDAD_FINANCIERA]
```

```
CREATE TABLE [dbo].[ENTIDAD_FINANCIERA](  
[IdEF] [char](5) NOT NULL,  
[RazonSocial] [nvarchar](60) NOT NULL,  
[Ruc] [char](11) NOT NULL,  
[Direccion] [nvarchar](100) NULL,  
[Email] [nvarchar](45) NULL,  
[Estado] [char](1) NOT NULL,  
[UserReg] [char](5) NOT NULL,  
[FechReg] [datetime] NOT NULL,  
[IpReg] [varchar](15) NOT NULL,  
[UserMod] [char](5) NULL,  
[FechMod] [datetime] NULL,  
[IpMod] [varchar](15) NULL,
```

```
[MotivMod] [nvarchar](60) NULL,
[UserBaja] [char](5) NULL,
[FechBaja] [datetime] NULL,
[IpBaja] [varchar](15) NULL,
[MotivBaja] [nvarchar](60) NULL
--ESTRUCTURA DE LA TABLA : [ZONA]
CREATE TABLE [dbo].[ZONA](
    [IdZona] [char](3) NOT NULL,
    [Zona] [nvarchar](30) NOT NULL,
    [Estado] [char](1) NOT NULL
--ESTRUCTURA DE LA TABLA : [AREA]
CREATE TABLE [dbo].[AREA](
    [IdArea] [char](5) NOT NULL,
    [Area] [nvarchar](30) NOT NULL,
    [Estado] [char](1) NOT NULL
--ESTRUCTURA DE LA TABLA : [CAR ]
CREATE TABLE [dbo].[CAR ](
    [IdCar ] [char](5) NOT NULL,
    [Car ] [nvarchar](20) NOT NULL,
    [Estado] [char](1) NOT NULL
--ESTRUCTURA DE LA TABLA : [DEPARTAMENTO]
CREATE TABLE [dbo].[DEPARTAMENTO](
    [IdDepartamento] [char](6) NOT NULL,
    [NombreDepartamento] [nvarchar](30) NOT NULL,
    [NombreCapitalDepa] [nvarchar](30) NOT NULL
--ESTRUCTURA DE LA TABLA : [PROVINCIA]
CREATE TABLE [dbo].[PROVINCIA](
    [IdProvincia] [char](6) NOT NULL,
    [NombreProvincia] [nvarchar](30) NOT NULL,
    [NombreCapitaProv] [nvarchar](30) NOT NULL,
    [IdDepartamento] [char](6) NOT NULL
--ESTRUCTURA DE LA TABLA : [DISTRITO]
CREATE TABLE [dbo].[DISTRITO](
    [IdDistrito] [char](7) NOT NULL,
    [NombreDistrito] [nvarchar](30) NOT NULL,
    [IdProvincia] [char](6) NOT NULL
--ESTRUCTURA DE LA TABLA : [PERSONA]
CREATE TABLE [dbo].[PERSONA](
    [IdPersona] [char](7) NOT NULL,
    [ApellidoPat] [nvarchar](25) NOT NULL,
    [ApellidoMat] [nvarchar](25) NOT NULL,
    [Nombres] [nvarchar](40) NOT NULL,
    [Dni] [char](8) NOT NULL,
    [Sexo] [char](1) NOT NULL,
    [EstadoCivil] [char](2) NULL,
    [TelfCasa] [nvarchar](16) NULL,
    [TelfTrabajo] [nvarchar](16) NULL,
    [TelfCelular] [nvarchar](16) NULL,
    [TelfRefel] [nvarchar](16) NULL,
    [Direccion] [nvarchar](60) NOT NULL,
    [IdDistrito] [char](7) NOT NULL,
    [Estado] [char](1) NOT NULL,
    [UserReg] [char](5) NULL,
    [FechReg] [datetime] NOT NULL,
    [IpReg] [varchar](15) NOT NULL,
    [UserMod] [char](5) NULL,
    [FechMod] [datetime] NULL,
    [IpMod] [varchar](15) NULL,
    [MotivMod] [nvarchar](60) NULL,
```

```
[UserBaja] [char](5) NULL,  
[FechBaja] [datetime] NULL,  
[IpBaja] [varchar](15) NULL,  
[MotivBaja] [nvarchar](60) NULL  
--ESTRUCTURA DE LA TABLA : [EMPLEADO]  
CREATE TABLE [dbo].[EMPLEADO](  
    [IdEmpleado] [char](6) NOT NULL,  
    [IdPersona] [char](7) NOT NULL,  
    [FechaNac] [date] NOT NULL,  
    [Email] [nvarchar](45) NULL,  
    [IdZona] [char](3) NOT NULL,  
    [IdArea] [char](5) NOT NULL,  
    [IdCar ] [char](5) NOT NULL,  
    [Estado] [char](1) NOT NULL,  
    [UserReg] [char](5) NOT NULL,  
    [FechReg] [datetime] NOT NULL,  
    [IpReg] [varchar](15) NOT NULL,  
    [UserMod] [char](5) NULL,  
    [FechMod] [datetime] NULL,  
    [IpMod] [varchar](15) NULL,  
    [MotivMod] [nvarchar](60) NULL,  
    [UserBaja] [char](5) NULL,  
    [FechBaja] [datetime] NULL,  
    [IpBaja] [varchar](15) NULL,  
    [MotivBaja] [nvarchar](60) NULL  
--ESTRUCTURA DE LA TABLA : [REPRESENTANTE_LEGAL]  
CREATE TABLE [dbo].[REPRESENTANTE_LEGAL](  
    [IdRepresLegal] [char](6) NOT NULL,  
    [IdEmpleado] [char](6) NOT NULL,  
    [Ruc] [char](11) NOT NULL,  
    [PartidaElectron] [char](9) NOT NULL,  
    [FechInicio] [date] NOT NULL,  
    [FechFin] [date] NULL,  
    [Estado] [char](1) NOT NULL,  
    [UserReg] [char](5) NOT NULL,  
    [FechReg] [datetime] NOT NULL,  
    [IpReg] [varchar](15) NOT NULL,  
    [UserMod] [char](5) NULL,  
    [FechMod] [datetime] NULL,  
    [IpMod] [varchar](15) NULL,  
    [MotivMod] [nvarchar](60) NULL,  
    [UserBaja] [char](5) NULL,  
    [FechBaja] [datetime] NULL,  
    [IpBaja] [varchar](15) NULL,  
    [MotivBaja] [nvarchar](60) NULL  
--ESTRUCTURA DE LA TABLA : [PROFESIONAL_RESIDENTE]  
CREATE TABLE [dbo].[PROFESIONAL_RESIDENTE](  
    [IdPr] [char](5) NOT NULL,  
    [IdEmpleado] [char](6) NOT NULL,  
    [Profesion] [nvarchar](30) NOT NULL,  
    [NumColegiatura] [nvarchar](15) NOT NULL,  
    [Estado] [char](1) NOT NULL,  
    [UserReg] [char](5) NOT NULL,  
    [FechReg] [datetime] NOT NULL,  
    [IpReg] [varchar](15) NOT NULL,  
    [UserMod] [char](5) NULL,  
    [FechMod] [datetime] NULL,  
    [IpMod] [varchar](15) NULL,  
    [MotivMod] [nvarchar](60) NULL,
```

```
[UserBaja] [char](5) NULL,  
[FechBaja] [datetime] NULL,  
[IpBaja] [varchar](15) NULL,  
[MotivBaja] [nvarchar](60) NULL  
--ESTRUCTURA DE LA TABLA : [COPIA_LITERAL]  
CREATE TABLE [dbo].[COPIA_LITERAL](  
    [IdCopiaLiteral] [char](7) NOT NULL,  
    [NumPartidaE] [nvarchar](9) NOT NULL,  
    [Direccion] [nvarchar](120) NOT NULL,  
    [Manzana] [nvarchar](6) NULL,  
    [Lote] [nvarchar](6) NULL,  
    [SubLote] [nvarchar](6) NULL,  
    [Referencia] [nvarchar](30) NOT NULL,  
    [Numero] [nvarchar](5) NULL,  
    [Interior] [nvarchar](5) NULL,  
    [IdDistrito] [char](7) NOT NULL,  
    [AreaTerrenoTotal] [nvarchar](6) NOT NULL,  
    [MedFrente] [nvarchar](5) NOT NULL,  
    [MedFondo] [nvarchar](5) NOT NULL,  
    [MedIzquierda] [nvarchar](5) NOT NULL,  
    [MedDerecha] [nvarchar](5) NOT NULL,  
    [Observacion] [nvarchar](100) NULL,  
    [FechaEmision] [date] NOT NULL,  
    [UserReg] [char](5) NOT NULL,  
    [FechReg] [datetime] NOT NULL,  
    [IpReg] [varchar](15) NOT NULL,  
    [UserMod] [char](5) NULL,  
    [FechMod] [datetime] NULL,  
    [IpMod] [varchar](15) NULL,  
    [MotivMod] [nvarchar](60) NULL,  
    [UserBaja] [char](5) NULL,  
    [FechBaja] [datetime] NULL,  
    [IpBaja] [varchar](15) NULL,  
    [MotivBaja] [nvarchar](60) NULL  
--ESTRUCTURA DE LA TABLA : [CAPTADO]  
CREATE TABLE [dbo].[CAPTADO](  
    [IdCaptado] [char](7) NOT NULL,  
    [IdPersona] [char](7) NOT NULL,  
    [CargaFam] [char](7) NULL,  
    [ParentescoCargaF] [nvarchar](10) NULL,  
    [IdCopiaLiteral] [char](7) NOT NULL,  
    [EstadoCap] [nvarchar](15) NOT NULL,  
    [Observacion] [nvarchar](200) NULL,  
    [Estado] [char](1) NOT NULL,  
    [UserReg] [char](5) NOT NULL,  
    [FechReg] [datetime] NOT NULL,  
    [IpReg] [varchar](15) NOT NULL,  
    [UserMod] [char](5) NULL,  
    [FechMod] [datetime] NULL,  
    [IpMod] [varchar](15) NULL,  
    [MotivMod] [nvarchar](60) NULL,  
    [UserBaja] [char](5) NULL,  
    [FechBaja] [datetime] NULL,  
    [IpBaja] [varchar](15) NULL,  
    [Motivbaja] [nvarchar](60) NULL  
--ESTRUCTURA DE LA TABLA : [ENTIDAD_TECNICA]  
CREATE TABLE [dbo].[ENTIDAD_TECNICA](  
    [IdEt] [char](5) NOT NULL,  
    [RazonSocial] [nvarchar](50) NOT NULL,
```

```
[Codi Et] [nvarchar](20) NOT NULL,  
[RucEt] [char](11) NOT NULL,  
[DireccionEt] [nvarchar](120) NOT NULL,  
[IdDistritoEt] [char](7) NOT NULL,  
[RepresLegal] [char](6) NOT NULL,  
[Estado] [char](1) NOT NULL,  
[UserReg] [char](5) NOT NULL,  
[FechReg] [datetime] NOT NULL,  
[IpReg] [varchar](15) NOT NULL,  
[UserMod] [char](5) NULL,  
[FechMod] [datetime] NULL,  
[IpMod] [varchar](15) NULL,  
[MotivMod] [nvarchar](60) NULL,  
[UserBaja] [char](5) NULL,  
[FechBaja] [datetime] NULL,  
[IpBaja] [varchar](15) NULL,  
[MotivBaja] [nvarchar](60) NULL  
--ESTRUCTURA DE LA TABLA : [BENEFICIARIO]  
CREATE TABLE [dbo].[BENEFICIARIO](  
[IdBeneficiario] [char](7) NOT NULL,  
[IdCaptado] [char](7) NOT NULL,  
[IdEt] [char](5) NOT NULL,  
[IdPr] [char](5) NOT NULL,  
[NumContrato] [char](13) NOT NULL,  
[Modalidad] [nvarchar](30) NOT NULL,  
[AreaConstruir] [decimal](6, 2) NOT NULL,  
[TazacionFabrica] [decimal](6, 2) NOT NULL,  
[PresupuestoObra] [decimal](8, 2) NOT NULL,  
[ValorTerreno] [decimal](8, 2) NOT NULL,  
[ValorBono] [decimal](8, 2) NOT NULL,  
[Ahorro] [decimal](8, 2) NOT NULL,  
[TiempoEjecucion] [tinyint] NOT NULL,  
[FechaFirmaContrato] [date] NULL,  
[FechaPa Ahorro] [date] NULL,  
[FechaRegistFMV] [date] NULL,  
[Codi FMV] [nvarchar](24) NULL,  
[FechaCodFMV] [date] NULL,  
[EstadoBene] [nvarchar](25) NULL,  
[Observacion] [nvarchar](300) NULL,  
[Estado] [char](1) NOT NULL,  
[UserReg] [char](5) NOT NULL,  
[FechReg] [datetime] NOT NULL,  
[IpReg] [varchar](15) NOT NULL,  
[UserBaja] [char](5) NULL,  
[FechBaja] [datetime] NULL,  
[IpBaja] [varchar](15) NULL,  
[MotivBaja] [nvarchar](60) NULL  
--ESTRUCTURA DE LA TABLA : [HISTORIAL_MODIFICACIONES]  
CREATE TABLE [dbo].[HISTORIAL_MODIFICACIONES](  
[IdHistMod] [char](7) NOT NULL,  
[IdBeneficiario] [char](7) NOT NULL,  
[Usuario] [char](5) NOT NULL,  
[Fecha] [datetime] NOT NULL,  
[Ip] [varchar](15) NOT NULL,  
[Motivo] [nvarchar](12) NOT NULL,  
[Tipo] [char](1) NOT NULL,  
[Campo] [nvarchar](20) NULL,  
[Valor] [nvarchar](300) NULL  
--ESTRUCTURA DE LA TABLA : [CAB_CARTA_FIANZA]
```

```
CREATE TABLE [dbo].[CAB_CARTA_FIANZA](
    [IdCF] [char](7) NOT NULL,
    [CF] [nvarchar](8) NOT NULL,
    [IdZona] [char](3) NOT NULL,
    [nCfBono] [nvarchar](20) NOT NULL,
    [nCfAhorro] [nvarchar](20) NOT NULL,
    [IdEntFinanc] [char](5) NOT NULL,
    [PorcLinea] [decimal](5, 2) NOT NULL,
    [FechEnvioCf] [date] NULL,
    [FechRecepCf] [date] NULL,
    [RigeDesde] [date] NOT NULL,
    [RigeHasta] [date] NOT NULL,
    [RigeHastaRenov] [date] NULL,
    [Observacion] [nvarchar](300) NULL,
    [ContRenov] [tinyint] NOT NULL,
    [Estado] [char](1) NOT NULL,
    [UserReg] [char](5) NOT NULL,
    [FechReg] [datetime] NOT NULL,
    [IpReg] [varchar](15) NOT NULL,
    [UserMod] [char](5) NULL,
    [FechMod] [datetime] NULL,
    [IpMod] [varchar](15) NULL,
    [MotivMod] [nvarchar](60) NULL,
    [UserBaja] [char](5) NULL,
    [FechBaja] [datetime] NULL,
    [IpBaja] [varchar](15) NULL,
    [MotivBaja] [nvarchar](60) NULL
--ESTRUCTURA DE LA TABLA : [CONSTRUCCION]
CREATE TABLE [dbo].[CONSTRUCCION](
    [IdConstru] [char](7) NOT NULL,
    [IdCF] [char](7) NOT NULL,
    [FechApertura] [date] NULL,
    [FechTermino] [date] NULL,
    [FechLiberada] [date] NULL,
    [Comentario] [nvarchar](300) NULL,
    [Estado] [char](1) NOT NULL,
    [UserApert] [char](5) NULL,
    [IpApert] [varchar](15) NULL,
    [UserReg] [char](5) NOT NULL,
    [FechReg] [datetime] NOT NULL,
    [IpReg] [varchar](15) NOT NULL,
    [UserMod] [char](5) NULL,
    [FechMod] [datetime] NULL,
    [IpMod] [varchar](15) NULL,
    [MotivMod] [nvarchar](60) NULL,
    [UserBaja] [char](5) NULL,
    [FechBaja] [datetime] NULL,
    [IpBaja] [varchar](15) NULL,
    [MotivBaja] [nvarchar](60) NULL
--ESTRUCTURA DE LA TABLA : [DET_CONSTRUCCION]
CREATE TABLE [dbo].[DET_CONSTRUCCION](
    [IdDetConst] [char](7) NOT NULL,
    [IdConstru] [char](7) NOT NULL,
    [IdBeneficiario] [char](7) NOT NULL,
    [IdEmpleadoSupv] [char](6) NULL,
    [MaestroObra] [nvarchar](30) NULL,
    [%Obra] [decimal](5, 2) NULL,
    [Comentario] [nvarchar](300) NULL,
    [EstadoObra] [nvarchar](25) NULL,
```

```
[FechInicio] [date] NULL,  
[FechTermino] [date] NULL,  
[Estado] [char](1) NULL,  
[UserReg] [char](5) NULL,  
[FechReg] [datetime] NULL,  
[IpReg] [varchar](15) NULL,  
[UserMod] [char](5) NULL,  
[FechMod] [datetime] NULL,  
[IpMod] [varchar](15) NULL,  
[MotivMod] [nvarchar](60) NULL,  
[UserBaja] [char](5) NULL,  
[FechBaja] [datetime] NULL,  
[IpBaja] [varchar](15) NULL,  
[MotivBaja] [nvarchar](60) NULL  
--ESTRUCTURA DE LA TABLA : [DET_CARTA_FIANZA]  
CREATE TABLE [dbo].[DET_CARTA_FIANZA](  
[IdDetCf] [char](7) NOT NULL,  
[IdCf] [char](7) NOT NULL,  
[IdBeneficiario] [char](7) NOT NULL,  
[Estado] [char](1) NOT NULL,  
[UserReg] [char](5) NOT NULL,  
[FechReg] [datetime] NOT NULL,  
[IpReg] [varchar](15) NOT NULL,  
[UserMod] [char](5) NULL,  
[FechMod] [datetime] NULL,  
[IpMod] [varchar](15) NULL,  
[MotivMod] [nvarchar](60) NULL,  
[UserBaja] [char](5) NULL,  
[FechBaja] [datetime] NULL,  
[IpBaja] [varchar](15) NULL,  
[MotivBaja] [nvarchar](60) NULL
```

## II. Programación del Sistema

### CLASES PRINCIPALES

**Descripción :** Contiene la definición de las Clases de Datos.

**Contenido :**

- Persona.cs
- Captado.cs
- Beneficiario.cs
- CartaFianza.cs

- **Persona.cs**

```
namespace ConstructoraBol
{
    public class Persona
    {
        #region Variables
        private string IdPersona;
        private string UserMod;
        private string ApellidoPat;
        private string ApellidoMat;
        private string Nombres;
        private string Dni;
        private string Sexo;
        private string EstadoCivil;
        private string TelefCasa;
        private string TelefTrabajo;
        private string TelefCelular;
        private string TelfRefe;
        private string Direccion;
        private string IdDistrito;
        private string Estado;
        private string UserReg;
        private string IpReg;
        #endregion

        #region Propiedades
        public string IpReg1
        {
            get { return IpReg; }
            set { IpReg = value; }
        }
        public string UserMod1
        {
            get { return UserMod; }
            set { UserMod = value; }
        }

        private string ipMod1;

        public string IpMod1
        {
            get { return ipMod1; }
            set { ipMod1 = value; }
        }
        public string IdPersona1
        {
            get { return IdPersona; }
            set { IdPersona = value; }
        }
        public static object InsertarPersona(Persona p)
        {
            DBACCES cnn = new DBACCES();
            SqlCommand cmd = new SqlCommand("PROC_INSERT_PERSONA", cnn.conexion);
            cmd.CommandType = CommandType.StoredProcedure;
            cnn.abrir();
            cmd.Parameters.Add(new SqlParameter("@ApellidoPat", p.ApellidoPat1));
            cmd.Parameters.Add(new SqlParameter("@ApellidoMat", p.ApellidoMat1));
            cmd.Parameters.Add(new SqlParameter("@Nombres", p.Nombres1));
            cmd.Parameters.Add(new SqlParameter("@Dni", p.Dni1));
        }
    }
}
```

```
cmd.Parameters.Add(new SqlParameter("@Sexo", p.Sexo1));
cmd.Parameters.Add(new SqlParameter("@EstadoCivil", p.EstadoCivil1));
cmd.Parameters.Add(new SqlParameter("@TelefCasa", p.TelefCasa1));
cmd.Parameters.Add(new SqlParameter("@TelefTrabajo", p.TelefTrabajo1));
cmd.Parameters.Add(new SqlParameter("@TelefCelular", p.TelefCelular1));
cmd.Parameters.Add(new SqlParameter("@TelfRefe", p.TelfRefe1));
cmd.Parameters.Add(new SqlParameter("@Direccion", p.Direccion1));
cmd.Parameters.Add(new SqlParameter("@IdDistrito", p.IdDistrito1));

cmd.Parameters.Add(new SqlParameter("@UserReg", p.UserReg1));
cmd.Parameters.Add(new SqlParameter("@IpReg", p.IpReg1));
cmd.ExecuteNonQuery();
cnn.cerrar();
return null;
}
public static DataTable ConsultaInfoComplePerso(Persona p)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_INFOCOMPLET_PERSONA";
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@Dni", p.Dni1));
    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}
```

```
}  
public static DataTable ConsultaBasicaPersona(Persona p)  
{  
    DBACCES cnn = new DBACCES();  
    SqlCommand cmd = new SqlCommand();  
    SqlDataReader miReadr;  
    DataTable mitab = new DataTable();  
    cmd.Connection = cnn.conexion;  
    cmd.CommandType = CommandType.StoredProcedure;  
    cmd.CommandText = "PROC_CONSULTA_BASICA_PERSONA";  
    cnn.abrir();  
    cmd.Parameters.Add(new SqlParameter("@IdPersona", p.IdPersona1));  
    miReadr = cmd.ExecuteReader();  
    mitab.Load(miReadr, LoadOption.OverwriteChanges);  
    return mitab;  
}  
  
public static DataTable ConsultaTotalPersona(Persona p)  
{  
    DBACCES cnn = new DBACCES();  
    SqlCommand cmd = new SqlCommand();  
    SqlDataReader miReadr;  
    DataTable mitab = new DataTable();  
    cmd.Connection = cnn.conexion;  
    cmd.CommandType = CommandType.StoredProcedure;  
    cmd.CommandText = "PROC_CONSULTA_TOTAL_PERSONA";  
    cnn.abrir();  
    cmd.Parameters.Add(new SqlParameter("@IdPersona", p.IdPersona1));  
    miReadr = cmd.ExecuteReader();  
    mitab.Load(miReadr, LoadOption.OverwriteChanges);  
    return mitab;  
}  
}
```

## Captado.cs

```
namespace ConstructoraBol  
{  
    public class Captado  
    {  
        #region variable  
        private string IpMod;  
  
        public string IpMod1  
        {  
            get { return IpMod; }  
            set { IpMod = value; }  
        }  
        private string cargaFa;  
  
        public string CargaFa1  
        {  
            get { return cargaFa; }  
            set { cargaFa = value; }  
        }  
        private string fechMod;
```

```
public string fechMod1
{
    get { return fechMod; }
    set { fechMod = value; }
}
private string idPersona;

public string IdPersona1
{
    get { return idPersona; }
    set { idPersona = value; }
}
private string CargaFamiliar;

public string CargaFamiliar1
{
    get { return CargaFamiliar; }
    set { CargaFamiliar = value; }
}
private string ParestescoCarga;

public string ParestescoCarga1
{
    get { return ParestescoCarga; }
    set { ParestescoCarga = value; }
}
private string EstadoCaptado;

public string EstadoCaptado1
{
    get { return EstadoCaptado; }
    set { EstadoCaptado = value; }
}
private string Observacion;

public string Observacion1
{
    get { return Observacion; }
    set { Observacion = value; }
}
private string UserReg;

public string UserReg1
{
    get { return UserReg; }
    set { UserReg = value; }
}
private string IpReg;

public string IpReg1
{
    get { return IpReg; }
    set { IpReg = value; }
}

private string IdZona;

public string IdZona1
```

```
{
    get { return IdZona; }
    set { IdZona = value; }
}
private string IdCaptado;

public string IdCaptado1
{
    get { return IdCaptado; }
    set { IdCaptado = value; }
}
private string IdCopiaLiteral;

public string IdCopiaLiteral1
{
    get { return IdCopiaLiteral; }
    set { IdCopiaLiteral = value; }
}
private string EstadoCap;

public string EstadoCap1
{
    get { return EstadoCap; }
    set { EstadoCap = value; }
}
private string Dni;

public string Dni1
{
    get { return Dni; }
    set { Dni = value; }
}

private string Estado;

public string Estado1
{
    get { return Estado; }
    set { Estado = value; }
}
private string PcCap;

public string PcCap1
{
    get { return PcCap; }
    set { PcCap = value; }
}
private DateTime FechUltMod;

public DateTime FechUltMod1
{
    get { return FechUltMod; }
    set { FechUltMod = value; }
}
private string PcModi;

public string PcModi1
{
    get { return PcModi; }
    set { PcModi = value; }
}
```

```

    }
    private DateTime FechElimi;

    public DateTime FechElimi1
    {
        get { return FechElimi; }
        set { FechElimi = value; }
    }
    private string PcElimi;

    public string PcElimi1
    {
        get { return PcElimi; }
        set { PcElimi = value; }
    }
    #endregion
    private string semana;
    private string semana;

    public string Semana1
    {
        get { return semana; }
        set { semana = value; }
    }
    private string año;

    public string Año1
    {
        get { return año; }
        set { año = value; }
    }
}

public static DataTable buscarcapxdni(Captado i)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_LLENAR_CAPTA_BENE";
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@dni", i.Dni1));
    cmd.Parameters.Add(new SqlParameter("@Idusuario", i.UserReg1));
    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}

public static object insertar(Captado c, CopiaLiteral i)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand("PROC_INSERT_CAPTADO",
cnn.conexion);
    cmd.CommandType = CommandType.StoredProcedure;
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@idPersona", c.IdPersona1));
    cmd.Parameters.Add(new SqlParameter("@CargaFamiliar",
c.CargaFamiliar1));
    cmd.Parameters.Add(new SqlParameter("@ParestescoCarga",
c.ParestescoCarga1));
}

```

```

        cmd.Parameters.Add(new SqlParameter("@EstadoCaptado",
c.EstadoCap1));
        cmd.Parameters.Add(new SqlParameter("@Observacion",
c.Observacion1));
        cmd.Parameters.Add(new SqlParameter("@UserReg", c.UserReg1));
        cmd.Parameters.Add(new SqlParameter("@IpReg", c.IpReg1));
        //INSERTANDO COPIA LITERAL
        cmd.Parameters.Add(new SqlParameter("@NumPartidaE",
i.NumPartidaE1));
        cmd.Parameters.Add(new SqlParameter("@Direccion", i.Direccion1));
        cmd.Parameters.Add(new SqlParameter("@Manzana", i.Manzana1));
        cmd.Parameters.Add(new SqlParameter("@Lote", i.Lote1));
        cmd.Parameters.Add(new SqlParameter("@SubLote", i.SubLote1));
        cmd.Parameters.Add(new SqlParameter("@Referencia", i.Referencia1));
        cmd.Parameters.Add(new SqlParameter("@Numero", i.Numero1));
        cmd.Parameters.Add(new SqlParameter("@Interior", i.Interior1));
        cmd.Parameters.Add(new SqlParameter("@IdDistrito", i.IdDistrito1));
        cmd.Parameters.Add(new SqlParameter("@AreaTerrenoTotal",
i.AreaTerrenoTotal1));
        cmd.Parameters.Add(new SqlParameter("@MedFrente", i.MedFrente1));
        cmd.Parameters.Add(new SqlParameter("@MedFondo", i.MedFondo1));
        cmd.Parameters.Add(new SqlParameter("@MedIzquierda",
i.MedIzquierda1));
        cmd.Parameters.Add(new SqlParameter("@MedDerecha", i.MedDerecha1));
        cmd.Parameters.Add(new SqlParameter("@ObservacionCL",
i.ObservacionCL1));
        cmd.Parameters.Add(new SqlParameter("@FechaEmision",
i.FechaEmision1));
        cmd.ExecuteNonQuery();
        cnn.cerrar();
        return null;
    }
#endregion
#region nuevasconsultas
public static DataTable BUSCARXDNIPERSONA(Captado i)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_BUXDNI_PERSONA";
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@dni", i.Dni1));
    cmd.Parameters.Add(new SqlParameter("@Idusuario", i.UserReg1));
    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}

public static object ModificarCargaFamiliar(Captado c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand("PROC_UPDATE_CARGA_FAM",
cnn.conexion);
    cmd.CommandType = CommandType.StoredProcedure;
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@cargaFa", c.CargaFa1));
    cmd.Parameters.Add(new SqlParameter("@IdCaptado",
c.IdCaptado1));

```

```

cmd.Parameters.Add(new SqlParameter("@UserMod", c.PcModi1));

cmd.Parameters.Add(new SqlParameter("@IpMod", c.IpMod1));

cmd.ExecuteNonQuery();
cnn.cerrar();
return null;
}

public static DataTable ConsultaDeCaptadosCuadro(Captado i)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_CONSULTA_GENERAL";
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@idusuario",
i.UserReg1));
    cmd.Parameters.Add(new SqlParameter("@semana", i.Semana1));
    cmd.Parameters.Add(new SqlParameter("@año", i.Año1));

    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}

public static DataTable ConsultaDigital(Captado i)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_CONSULTA_DIGITAL";
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@idusuario",
i.UserReg1));
    cmd.Parameters.Add(new SqlParameter("@semana", i.Semana1));
    cmd.Parameters.Add(new SqlParameter("@año", i.Año1));

    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}

public static DataTable ConsultaDeCaptadosAño()
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_SELECT_AÑO";
    cnn.abrir();
    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
}

```

```
        return mitab;
    }

    public static DataTable ConsultaSemanaConFiltro(Captado i)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand();
        SqlDataReader miReadr;
        DataTable mitab = new DataTable();
        cmd.Connection = cnn.conexion;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "PROC_SELECT_SEMANA_FILTRO";
        cmd.Parameters.Add(new SqlParameter("@año", i.Año1));
        cnn.abrir();
        miReadr = cmd.ExecuteReader();
        mitab.Load(miReadr, LoadOption.OverwriteChanges);
        return mitab;
    }

    public static DataTable ConsultaDeZona(Captado i)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand();
        SqlDataReader miReadr;
        DataTable mitab = new DataTable();
        cmd.Connection = cnn.conexion;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "PROC_SELECT_ZONA_FILTRO";

        cnn.abrir();
        miReadr = cmd.ExecuteReader();
        mitab.Load(miReadr, LoadOption.OverwriteChanges);
        return mitab;
    }

    public static DataTable ConsultaCaptadorPorZona(Captado i)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand();
        SqlDataReader miReadr;
        DataTable mitab = new DataTable();
        cmd.Connection = cnn.conexion;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "PROC_SELECT_USUARIO_POR_ZONA";
        cmd.Parameters.Add(new SqlParameter("@IdZona",
i.IdZona1));

        cnn.abrir();
        miReadr = cmd.ExecuteReader();
        mitab.Load(miReadr, LoadOption.OverwriteChanges);
        return mitab;
    }

    public static DataTable ConsultaDigitalIII(Captado i)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand();
```

```
        SqlDataReader miReadr;  
        DataTable mitab = new DataTable();  
        cmd.Connection = cnn.conexion;  
        cmd.CommandType = CommandType.StoredProcedure;  
        cmd.CommandText = "PROC_CONSULTA_DIGITALII";  
        cnn.abrir();  
        cmd.Parameters.Add(new SqlParameter("@IdZona",  
i.IdZona1));  
        cmd.Parameters.Add(new SqlParameter("@idusuario",  
.UserReg1));  
        cmd.Parameters.Add(new SqlParameter("@semana",  
i.Semana1));  
        cmd.Parameters.Add(new SqlParameter("@año", i.Año1));  
  
        miReadr = cmd.ExecuteReader();  
        mitab.Load(miReadr, LoadOption.OverwriteChanges);  
        return mitab;  
    }  
}
```

## BENEFICIARIO.CS

```
namespace ConstructoraBol  
{  
    public class Beneficiario  
    {  
        #region Variables  
        private string cargaFa;  
  
        public string CargaFa1  
        {  
            get { return cargaFa; }  
            set { cargaFa = value; }  
        }  
        private string IdBeneficiario;  
  
        public string IdBeneficiario1  
        {  
            get { return IdBeneficiario; }  
            set { IdBeneficiario = value; }  
        }  
        private string IdCaptado;  
  
        public string IdCaptado1  
        {  
            get { return IdCaptado; }  
            set { IdCaptado = value; }  
        }  
        private string IdEt;  
  
        public string IdEt1  
        {  
            get { return IdEt; }  
            set { IdEt = value; }  
        }  
        private string IdPr;  
  
        public string IdPr1
```

```
{
    get { return IdPr; }
    set { IdPr = value; }
}
private string NumContrato;

public string NumContrato1
{
    get { return NumContrato; }
    set { NumContrato = value; }
}
private string Modalidad;

public string Modalidad1
{
    get { return Modalidad; }
    set { Modalidad = value; }
}
private decimal AreaConstruir;

public decimal AreaConstruir1
{
    get { return AreaConstruir; }
    set { AreaConstruir = value; }
}
private decimal TazacionFabrica;

public decimal TazacionFabrica1
{
    get { return TazacionFabrica; }
    set { TazacionFabrica = value; }
}
private decimal PresupuestoObra;

public decimal PresupuestoObra1
{
    get { return PresupuestoObra; }
    set { PresupuestoObra = value; }
}
private decimal ValorTerreno;

public decimal ValorTerreno1
{
    get { return ValorTerreno; }
    set { ValorTerreno = value; }
}
private decimal ValorBono;

public decimal ValorBono1
{
    get { return ValorBono; }
    set { ValorBono = value; }
}
private decimal Ahorro;

public decimal Ahorro1
{
    get { return Ahorro; }
    set { Ahorro = value; }
}
```

```
private DateTime FechaFirmaContrato;

public DateTime FechaFirmaContrato1
{
    get { return FechaFirmaContrato; }
    set { FechaFirmaContrato = value; }
}
private int TiempoEjecucion;

public int TiempoEjecucion1
{
    get { return TiempoEjecucion; }
    set { TiempoEjecucion = value; }
}
private DateTime FechaRegistFMV;

public DateTime FechaRegistFMV1
{
    get { return FechaRegistFMV; }
    set { FechaRegistFMV = value; }
}
private string OBSERVACION;

public string OBSERVACION1
{
    get { return OBSERVACION; }
    set { OBSERVACION = value; }
}

private string idUsuario;

public string idUsuario1
{
    get { return idUsuario; }
    set { idUsuario = value; }
}
private string IdZona;

public string IdZona1
{
    get { return IdZona; }
    set { IdZona = value; }
}
private string dni;

public string Dni1
{
    get { return dni; }
    set { dni = value; }
}
private string UserModi;

public string UserModi1
{
    get { return UserModi; }
    set { UserModi = value; }
}
private stringCodigoFmv;

public stringCodigoFmv1
```

```
{
    get { returnCodigoFmv; }
    set {CodigoFmv = value; }
}
private DateTime FechaCodFMV;

public DateTime FechaCodFMV1
{
    get { return FechaCodFMV; }
    set { FechaCodFMV = value; }
}
private string UserCodi;

public string UserCodi1
{
    get { return UserCodi; }
    set { UserCodi = value; }
}

private string Usuario;

public string Usuario1
{
    get { return Usuario; }
    set { Usuario = value; }
}
private DateTime FechaPagoAhorro;

public DateTime FechaPagoAhorro1
{
    get { return FechaPagoAhorro; }
    set { FechaPagoAhorro = value; }
}
private string IpReg;

public string IpReg1
{
    get { return IpReg; }
    set { IpReg = value; }
}
private string MotivoReg;

public string MotivoReg1
{
    get { return MotivoReg; }
    set { MotivoReg = value; }
}
private string Tipo;

public string Tipo1
{
    get { return Tipo; }
    set { Tipo = value; }
}
private string Campo;

public string Campo1
{
    get { return Campo; }
    set { Campo = value; }
}
```

```

    }
    private string Valor;

    public string Valor1
    {
        get { return Valor; }
        set { Valor = value; }
    }

    #endregion

    public static DataTable Consultarcontrato(Beneficiario i)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand();
        SqlDataReader miReadr;
        DataTable mitab = new DataTable();
        cmd.Connection = cnn.conexion;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "MOSTRARNUMEROCONTRATO";
        cnn.abrir();
        cmd.Parameters.Add(new SqlParameter("@IdCAPTADO", i.IdCaptado1));
        miReadr = cmd.ExecuteReader();
        mitab.Load(miReadr, LoadOption.OverwriteChanges);
        return mitab;
    }

    public static DataTable Seguimiento(Persona i)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand();
        SqlDataReader miReadr;
        DataTable mitab = new DataTable();
        cmd.Connection = cnn.conexion;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "Proc_SeguimientoClientes";
        cnn.abrir();
        cmd.Parameters.Add(new SqlParameter("@Dni", i.Dni1));
        miReadr = cmd.ExecuteReader();
        mitab.Load(miReadr, LoadOption.OverwriteChanges);
        return mitab;
    }

    public static object InsertarBene(Beneficiario c)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new
        SqlCommand("GUARDAR_BENE_GENERAR_NUM_CONTRATO", cnn.conexion);
        cmd.CommandType = CommandType.StoredProcedure;
        cnn.abrir();
        cmd.Parameters.Add(new SqlParameter("@IdCaptado", c.IdCaptado1));
        cmd.Parameters.Add(new SqlParameter("@IdEt", c.IdEt1));
        cmd.Parameters.Add(new SqlParameter("@IdPr", c.IdPr1));
        cmd.Parameters.Add(new SqlParameter("@Modalidad", c.Modalidad1));
        cmd.Parameters.Add(new SqlParameter("@AreaConstruir",
        c.AreaConstruir1));
        cmd.Parameters.Add(new SqlParameter("@TazacionFabrica",
        c.TazacionFabrica1));
        cmd.Parameters.Add(new SqlParameter("@PresupuestoObra",
        c.PresupuestoObra1));
        cmd.Parameters.Add(new SqlParameter("@ValorTerreno",

```

```

c.ValorTerreno1));
    cmd.Parameters.Add(new SqlParameter("@ValorBono", c.ValorBono1));
    cmd.Parameters.Add(new SqlParameter("@Ahorro", c.Ahorro1));
    cmd.Parameters.Add(new SqlParameter("@TiempoEjecucion",
c.TiempoEjecucion1));
    cmd.Parameters.Add(new SqlParameter("@Observacion",
c.OBSERVACION1));
    cmd.Parameters.Add(new SqlParameter("@UserReg", c.idUsuario1));
    cmd.Parameters.Add(new SqlParameter("@IdZona", c.IdZona1));
    cmd.Parameters.Add(new SqlParameter("@IpReg", c.IpReg1));
    cmd.ExecuteNonQuery();
    cnn.cerrar();
    return null;
}
public static object AgregarModificacion(Beneficiario c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand("PROC_AGREGAR_MODIFICACIONES",
cnn.conexion);
    cmd.CommandType = CommandType.StoredProcedure;
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@UserReg", c.idUsuario1));
    cmd.Parameters.Add(new SqlParameter("@Ip", c.IpReg1));
    cmd.Parameters.Add(new SqlParameter("@IdBeneficiario",
c.IdBeneficiario1));
    cmd.Parameters.Add(new SqlParameter("@MotivoReg", c.MotivoReg1));
    cmd.Parameters.Add(new SqlParameter("@Tipo", c.Tipo1));
    cmd.Parameters.Add(new SqlParameter("@Campo", c.Campo1));
    cmd.Parameters.Add(new SqlParameter("@Valor", c.Valor1));
    cmd.ExecuteNonQuery();
    cnn.cerrar();
    return null;
}
public static object ModificarBeneficiarioPrin(Beneficiario c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand("PROC_UPDATE_BENE_PRINC",
cnn.conexion);
    cmd.CommandType = CommandType.StoredProcedure;
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@IdBeneficiario",
c.IdBeneficiario1));
    cmd.Parameters.Add(new SqlParameter("@IdEt", c.IdEt1));
    cmd.Parameters.Add(new SqlParameter("@IdPr", c.IdPr1));
    cmd.Parameters.Add(new SqlParameter("@Modalidad", c.Modalidad1));
    cmd.Parameters.Add(new SqlParameter("@AreaConstruir",
c.AreaConstruir1));
    cmd.Parameters.Add(new SqlParameter("@TazacionFabrica",
c.TazacionFabrica1));
    cmd.Parameters.Add(new SqlParameter("@PresupuestoObra",
c.PresupuestoObra1));
    cmd.Parameters.Add(new SqlParameter("@ValorTerreno",
c.ValorTerreno1));
    cmd.Parameters.Add(new SqlParameter("@ValorBono", c.ValorBono1));
    cmd.Parameters.Add(new SqlParameter("@Ahorro", c.Ahorro1));
    cmd.Parameters.Add(new SqlParameter("@TiempoEjecucion",
c.TiempoEjecucion1));
    cmd.Parameters.Add(new SqlParameter("@Observacion",
c.OBSERVACION1));
    cmd.ExecuteNonQuery();
}

```

```

        cnn.cerrar();
        return null;
    }
    public static DataTable BuscarBeneCodigo(Beneficiario i)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand();
        SqlDataReader miReadr;
        DataTable mitab = new DataTable();
        cmd.Connection = cnn.conexion;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "PROC_BUSCAR_BENECODIGO";
        cnn.abrir();
        cmd.Parameters.Add(new SqlParameter("@Dni", i.Dni1));
        cmd.Parameters.Add(new SqlParameter("@Idusuario", i.idUsuario1));
        miReadr = cmd.ExecuteReader();
        mitab.Load(miReadr, LoadOption.OverwriteChanges);
        return mitab;
    }
    public static DataTable BuscarBenePago(Beneficiario i)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand();
        SqlDataReader miReadr;
        DataTable mitab = new DataTable();
        cmd.Connection = cnn.conexion;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "PROC_BUXDNI_AHORRO";
        cnn.abrir();
        cmd.Parameters.Add(new SqlParameter("@Dni", i.Dni1));
        cmd.Parameters.Add(new SqlParameter("@Idusuario", i.idUsuario1));
        miReadr = cmd.ExecuteReader();
        mitab.Load(miReadr, LoadOption.OverwriteChanges);
        return mitab;
    }
    public static DataTable buscarBENE(Beneficiario i)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand();
        SqlDataReader miReadr;
        DataTable mitab = new DataTable();
        cmd.Connection = cnn.conexion;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "buscarbene1";
        cnn.abrir();
        cmd.Parameters.Add(new SqlParameter("@Dni", i.Dni1));
        cmd.Parameters.Add(new SqlParameter("@IdZona", i.IdZona1));
        miReadr = cmd.ExecuteReader();
        mitab.Load(miReadr, LoadOption.OverwriteChanges);
        return mitab;
    }
    public static object ModificarFechaContrato(Beneficiario c)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new
        SqlCommand("PROC_UPDATE_FECHA_FIRMA_CONTRATO", cnn.conexion);
        cmd.CommandType = CommandType.StoredProcedure;
        cnn.abrir();
        cmd.Parameters.Add(new SqlParameter("@UserReg", c.idUsuario1));
        cmd.Parameters.Add(new SqlParameter("@Ip", c.IpReg1));
    }

```

```

        cmd.Parameters.Add(new SqlParameter("@IdBeneficiario",
c.IdBeneficiario1));
        cmd.Parameters.Add(new SqlParameter("@FechaFirmaContrato",
c.FechaFirmaContrato1));
        cmd.Parameters.Add(new SqlParameter("@MotivoReg",
c.MotivoReg1));
        cmd.Parameters.Add(new SqlParameter("@Tipo", c.Tipo1));
        cmd.Parameters.Add(new SqlParameter("@Campo", c.Campo1));
        cmd.Parameters.Add(new SqlParameter("@Valor", c.Valor1));
        cmd.ExecuteNonQuery();
        cnn.cerrar();
        return null;
    }
    public static object ModificarFechaPago(Beneficiario b)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand("PROC_UPDATE_FECHAHORRO",
cnn.conexion);
        cmd.CommandType = CommandType.StoredProcedure;
        cnn.abrir();
        cmd.Parameters.Add(new SqlParameter("@IdBeneficiario",
b.IdBeneficiario1));
        cmd.Parameters.Add(new SqlParameter("@FechaPagoAhorro",
b.FechaPagoAhorro1));
        cmd.Parameters.Add(new SqlParameter("@UserReg", b.idUsuario1));
        cmd.Parameters.Add(new SqlParameter("@Ip", b.IpReg1));
        cmd.Parameters.Add(new SqlParameter("@MotivoReg",
b.MotivoReg1));
        cmd.Parameters.Add(new SqlParameter("@Tipo", b.Tipo1));
        cmd.Parameters.Add(new SqlParameter("@Campo", b.Campo1));
        cmd.Parameters.Add(new SqlParameter("@Valor", b.Valor1));
        cmd.ExecuteNonQuery();
        cnn.cerrar();
        return null;
    }
    public static object ModificarFondoMV(Beneficiario c)
    {
        DBACCES cnn = new DBACCES();
        SqlCommand cmd = new SqlCommand("PROC_UPDATE_FECHA_FMV",
cnn.conexion);
        cmd.CommandType = CommandType.StoredProcedure;
        cnn.abrir();
        cmd.Parameters.Add(new SqlParameter("@UserReg", c.idUsuario1));
        cmd.Parameters.Add(new SqlParameter("@Ip", c.IpReg1));
        cmd.Parameters.Add(new SqlParameter("@IdBeneficiario",
c.IdBeneficiario1));
        cmd.Parameters.Add(new SqlParameter("@FechaRegistroFMV",
c.FechaRegistFMV1));
        cmd.Parameters.Add(new SqlParameter("@MotivoReg",
c.MotivoReg1));
        cmd.Parameters.Add(new SqlParameter("@Tipo", c.Tipo1));
        cmd.Parameters.Add(new SqlParameter("@Campo", c.Campo1));
        cmd.Parameters.Add(new SqlParameter("@Valor", c.Valor1));
        cmd.ExecuteNonQuery();
        cnn.cerrar();
        return null;
    }
    public static DataTable SelecBenefCf(Beneficiario i)

```

```
        {
            DBACCES cnn = new DBACCES();
            SqlCommand cmd = new SqlCommand();
            SqlDataReader miReadr;
            DataTable mitab = new DataTable();
            cmd.Connection = cnn.conexion;
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.CommandText = "PROC_CONSULT_BENXDNI_CF";
            cnn.abrir();
            cmd.Parameters.Add(new SqlParameter("@Dni", i.Dni1));
            miReadr = cmd.ExecuteReader();
            mitab.Load(miReadr, LoadOption.OverwriteChanges);
            return mitab;
        }
    }
}

CARTA FIANZA.CS
namespace ConstructoraBol
{
    public class CartaFianza
    {
        #region variables y propiedades

        private string IdCF;

        public string IdCF1
        {
            get { return IdCF; }
            set { IdCF = value; }
        }

        private string CF;

        public string CF1
        {
            get { return CF; }
            set { CF = value; }
        }

        private string IdZona;

        public string IdZona1
        {
            get { return IdZona; }
            set { IdZona = value; }
        }

        private string nCfBono;

        public string NCfBono
        {
            get { return nCfBono; }
            set { nCfBono = value; }
        }

        private string nCfAhorro;

        public string NCfAhorro
        {
            get { return nCfAhorro; }
            set { nCfAhorro = value; }
        }

        private string IdEntFinanc;

        public string IdEntFinanc1
        {

```

```
        get { return IdEntFinanc; }
        set { IdEntFinanc = value; }
    }
    private decimal PorcLinea;

    public decimal PorcLinea1
    {
        get { return PorcLinea; }
        set { PorcLinea = value; }
    }
    private DateTime FechEnvioCf;

    public DateTime FechEnvioCf1
    {
        get { return FechEnvioCf; }
        set { FechEnvioCf = value; }
    }
    private DateTime FechRecepCf;

    public DateTime FechRecepCf1
    {
        get { return FechRecepCf; }
        set { FechRecepCf = value; }
    }
    private DateTime RigeDesde;

    public DateTime RigeDesde1
    {
        get { return RigeDesde; }
        set { RigeDesde = value; }
    }
    private DateTime RigeHasta;

    public DateTime RigeHasta1
    {
        get { return RigeHasta; }
        set { RigeHasta = value; }
    }
    private DateTime RigeHastaRenov;

    public DateTime RigeHastaRenov1
    {
        get { return RigeHastaRenov; }
        set { RigeHastaRenov = value; }
    }
    private string Observacion;

    public string Observacion1
    {
        get { return Observacion; }
        set { Observacion = value; }
    }
    private Int16 ContRenov;

    public Int16 ContRenov1
    {
        get { return ContRenov; }
        set { ContRenov = value; }
    }
    private string User;
```

```

public string User1
{
    get { return User; }
    set { User = value; }
}
private string Ip;

public string Ip1
{
    get { return Ip; }
    set { Ip = value; }
}
private string Motiv;

public string Motiv1
{
    get { return Motiv; }
    set { Motiv = value; }
}

private string IdDetCf;

public string IdDetCf1
{
    get { return IdDetCf; }
    set { IdDetCf = value; }
}

private string IdBeneficiario;

public string IdBeneficiario1
{
    get { return IdBeneficiario; }
    set { IdBeneficiario = value; }
}

#endregion

public static object InsertarCF(CartaFianza c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand("PROC_INSERTAR_CF", cnn.conexion);
    cmd.CommandType = CommandType.StoredProcedure;
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@CF", c.CF1));
    cmd.Parameters.Add(new SqlParameter("@IdZona", c.IdZona1));
    cmd.Parameters.Add(new SqlParameter("@nCfBono", c.NCfBono));
    cmd.Parameters.Add(new SqlParameter("@nCfAhorro", c.NCfAhorro));
    cmd.Parameters.Add(new SqlParameter("@IdEntFinanc",
c.IdEntFinanc1));
    cmd.Parameters.Add(new SqlParameter("@PorcLinea", c.PorcLinea1));
    cmd.Parameters.Add(new SqlParameter("@FechEnvioCf",
c.FechEnvioCf1));
    cmd.Parameters.Add(new SqlParameter("@FechRecepCf",
c.FechRecepCf1));
    cmd.Parameters.Add(new SqlParameter("@RigeDesde", c.RigeDesde1));
    cmd.Parameters.Add(new SqlParameter("@RigeHasta", c.RigeHasta1));
    cmd.Parameters.Add(new SqlParameter("@Observacion",
c.Observacion1));
}

```

```

cmd.Parameters.Add(new SqlParameter("@UserReg", c.User1));
cmd.Parameters.Add(new SqlParameter("@IpReg", c.Ip1));
cmd.ExecuteNonQuery();
cnn.cerrar();
return null;
}
public static DataTable ObtenerIdCF(CartaFianza c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_OBTENER_IDCF";
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@CF", c.CF1));
    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}
public static object InsertarBenefCF(CartaFianza c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand("PROC_INSERTAR_BENEF_CF",
cnn.conexion);
    cmd.CommandType = CommandType.StoredProcedure;
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@IdCf", c.IdCF1));
    cmd.Parameters.Add(new SqlParameter("@IdBeneficiario",
c.IdBeneficiario1));
    cmd.Parameters.Add(new SqlParameter("@UserReg", c.User1));
    cmd.Parameters.Add(new SqlParameter("@IpReg", c.Ip1));
    cmd.ExecuteNonQuery();
    cnn.cerrar();
    return null;
}
public static DataTable consultaCF(CartaFianza c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_CONSULTA_CF";
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@CF", c.CF1));
    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}
public static object RenovarCF(CartaFianza c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand("PROC_RENOVAR_CF", cnn.conexion);
    cmd.CommandType = CommandType.StoredProcedure;
    cnn.abrir();

```

```
cmd.Parameters.Add(new SqlParameter("@IdCF", c.IdCF1));
cmd.Parameters.Add(new SqlParameter("@RigeHastaRenov",
c.RigeHastaRenov1));
cmd.Parameters.Add(new SqlParameter("@UserMod", c.User1));
cmd.Parameters.Add(new SqlParameter("@IpMod", c.Ip1));
cmd.ExecuteNonQuery();
cnn.cerrar();
return null;
}
public static DataTable consultaCF_Apertura()
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_SELECT_CF_APERTURAR";
    cnn.abrir();
    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}
public static DataTable consultaCF_AperturaFiltro(CartaFianza c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_SELECT_FILTRO_CF_APERTURAR";
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@CF", c.CF1));
    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}
public static DataTable consultarMunicipalidadesCF(CartaFianza c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
    SqlDataReader miReadr;
    DataTable mitab = new DataTable();
    cmd.Connection = cnn.conexion;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "PROC_MUNICIPALIDADES_CF_APERT";
    cnn.abrir();
    cmd.Parameters.Add(new SqlParameter("@IdCF", c.IdCF1));
    miReadr = cmd.ExecuteReader();
    mitab.Load(miReadr, LoadOption.OverwriteChanges);
    return mitab;
}
public static DataTable consultarBeneficiariosCF(CartaFianza c)
{
    DBACCES cnn = new DBACCES();
    SqlCommand cmd = new SqlCommand();
```

```
SqlDataReader miReadr;  
DataTable mitab = new DataTable();  
cmd.Connection = cnn.conexion;  
cmd.CommandType = CommandType.StoredProcedure;  
cmd.CommandText = "PROC_MOSTRAR_BENEF_CF_APERT";  
cnn.abrir();  
cmd.Parameters.Add(new SqlParameter("@IdCF", c.IdCF1));  
miReadr = cmd.ExecuteReader();  
mitab.Load(miReadr, LoadOption.OverwriteChanges);  
return mitab;  
  
}
```

## PÁGINAS WEB PRINCIPALES

**Descripción :** Contiene las Páginas Web de la Aplicación  
**Contenido :**

- AgregarPersona.aspx.cs
- AgregarCaptado.aspx.cs
- Beneficiario.aspx.cs
- AperturarCartaFianza.aspx.cs
- ActualizacionFMV.aspx.cs
- ActualizacionFechaFirmaContrato.aspx.cs

```
Agregar Persona.aspx.cs  
namespace Constructora.Sistema  
{  
    public partial class AgregarPersona : System.Web.UI.Page  
    {  
        protected void Page_Load(object sender, EventArgs e)  
        {  
            if (Variables.Pcaptado != "A")  
            {  
                Response.Redirect("~/AccesoDenegado.aspx");  
            }  
            if (!IsPostBack)  
            {  
                llenar_Cmb(cbDepa, "DEP");  
            }  
        }  
  
        void llenar_Cmb(DropDownList cmb, String val)  
        {  
            Departamento ubi = new Departamento();  
            switch (val)  
            {  
  
                case "DEP":
```

```

        cmb.DataSource = Departamento.listar_Dep_Pro_Dis(ubi,
"DEP");

        cmb.DataTextField = "NombreDepartamento";
        cmb.DataValueField = "IdDepartamento";

        cmb.DataBind();
        cmb.Items.Insert(0, new ListItem("Seleccione
Departamento...", ""));
        break;
        case "PRO":
            ubi.IdDepartamento1 =
Convert.ToString(cbDepa.SelectedValue);
            cmb.DataSource = Departamento.listar_Dep_Pro_Dis(ubi,
"PRO");

            cmb.DataTextField = "NombreProvincia";
            cmb.DataValueField = "IdProvincia";
            cmb.DataBind();
            cmb.Items.Insert(0, new ListItem("Seleccione Provincia...",
""));

            break;
        case "DIS":
            ubi.IdProvincia1 = Convert.ToString(CbProv.SelectedValue);
            cmb.DataSource = Departamento.listar_Dep_Pro_Dis(ubi,
"DIS");

            cmb.DataTextField = "NombreDistrito";
            cmb.DataValueField = "IdDistrito";
            cmb.DataBind();
            cmb.Items.Insert(0, new ListItem("Seleccione Distrito...",
""));

            break;
    }
}

protected void CbProv_SelectedIndexChanged(object sender, EventArgs e)
{
    llenar_Cmb(cbdistrit, "DIS");
}
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        ConstructoraBol.Persona pe = new ConstructoraBol.Persona();
        pe.ApellidoPat1 = txtapellido0.Text.ToUpper();
        pe.ApellidoMat1 = txtMaterno0.Text.ToUpper();
        pe.Nombres1 = txtNombre0.Text.ToUpper();
        pe.Dni1 = txtdni0.Text;
        if (rbfemenino.Checked == true)
        {
            pe.Sexo1 = "F";
        }
        if (rbhombre.Checked == true)
        {
            pe.Sexo1 = "M";
        }
        pe.EstadoCivil1 = ddlcivil0.Text;
        pe.TelefCasa1 = txttelfcasa.Text;
        pe.TelefTrabajo1 = txttlftrabajo.Text;
        pe.TelefCelular1 = txttlfcelular.Text;
    }
}

```

```

pe.TelfRef1 = txttlfreferencia.Text.ToUpper();
pe.Direccion1 = txtdireccion.Text.ToUpper();
if (cbdistrib.Text == "Seleccione Distrito..." || cbdistrib.Text
== "")
{
    string script3 = @"<script type='text/javascript'>
alert('SE NECESITA INGRESAR EL UBIGEO');
</script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script3, false);
}
else
{
    pe.IdDistrito1 = Convert.ToString(cbdistrib.SelectedValue);
}
//estado
//PE.UserReg1=Variables.idusuario;
pe.UserReg1 = Variables.idusuario;
pe.IpReg1 = Variables.IpPublic;
ConstructoraBol.Persona.InsertarPersona(pe);
string mensaje = "Ah registrado a persona : "
+txtapellido0.Text.ToUpper() + " " + txtMaterno0.Text + " " + txtNombre0.Text + "
" + "Con el DNI número : " + txtdni0.Text;
string script1 = @"<script type='text/javascript'>alert(' " +
mensaje + "');</script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script1, false);
}
catch (Exception c)
{
    string script1 = @"<script type='text/javascript'>
alert('VERIFIQUE DNI ES POSIBLE QUE YA EXISTA');
</script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script1, false);
}
}
}

```

### AgregarCaptado.aspx.cs

```

namespace Constructora.Sistema
{
    public partial class AgregarCaptado : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Variables.Pcaptado != "A")
            {
                Response.Redirect("~/AccesoDenegado.aspx");
            }
            if (!IsPostBack)
            {
                Button2.Enabled = false;
                txtfechaEmi.Text = DateTime.Now.Date.ToString("d/MM/yyyy");

                llenar_Cmb(cbDepa, "DEP");

                fechaVencimiento();
            }
        }
    }
}

```

```
    }
}
protected void cbDepa_SelectedIndexChanged(object sender, EventArgs e)
{
    llenar_Cmb(CbProv, "PRO");
}
protected void CbProv_SelectedIndexChanged(object sender, EventArgs e)
{
    llenar_Cmb(cbdistrit, "DIS");
}

protected void Button1_Click(object sender, EventArgs e)
{
    ConstructoraBol.Captado Cap = new ConstructoraBol.Captado();
    Cap.Dni1 = txtDnititu.Text;
    Cap.UserReg1 = Variables.idusuario;
    DataTable obj = ConstructoraBol.Captado.BUSCARXDNIPERSONA(Cap);
    lblpersona.Text = obj.Rows[0][0].ToString();
    txtapPT.Text = obj.Rows[0][1].ToString();
    txtapMt.Text = obj.Rows[0][2].ToString();
    txtNoTi.Text = obj.Rows[0][3].ToString();
    string sexo = obj.Rows[0][4].ToString();
    if (sexo == "M")
    {
        txtSexoTitu.Text = "MASCULINO";
    }
    else if (sexo == "F")
    {
        txtSexoTitu.Text = "FEMENINO";
    }
    string estadocivil = obj.Rows[0][5].ToString();
    switch (estadocivil)
    {
        case "CA":
        {
            txtestadoCivil.Text = "CASADO";
            break;
        }
        case "SO":
        {
            txtestadoCivil.Text = "SOLTERO";
            break;
        }
        case "VI":
        {
            txtestadoCivil.Text = "VIUDO";
            break;
        }
        case "DI":
        {
            txtestadoCivil.Text = "DIVORCIO";
            break;
        }
    }

    txttelCasaT.Text = obj.Rows[0][6].ToString();
    txtTeleTrab.Text = obj.Rows[0][7].ToString();
    txtTeleCelu.Text = obj.Rows[0][8].ToString();
}
```

```

txtTelRefe.Text = obj.Rows[0][9].ToString();
lbldni.Text = txtdniCF.Text;
Button2.Enabled = true;
}

protected void Button2_Click(object sender, EventArgs e)
{
    if (lbldni.Text != txtdniCF.Text)
    {
        ConstructoraBol.Captado Cap = new ConstructoraBol.Captado();
        Cap.Dni1 = txtdniCF.Text;
        Cap.UserReg1 = Variables.idusuario;
        DataTable ObjCF =
ConstructoraBol.Captado.BUSCARXDNIPERSONA(Cap);
        lblcargafamiliar.Text = ObjCF.Rows[0][0].ToString();
        txtapPatCF.Text = ObjCF.Rows[0][1].ToString();
        txtApCF.Text = ObjCF.Rows[0][2].ToString();
        txtNombreCF.Text = ObjCF.Rows[0][3].ToString();
        string sexo = ObjCF.Rows[0][4].ToString();
        if (sexo == "M")
        {
            txtsexoCF.Text = "MASCULINO";
        }
        else if (sexo == "F")
        {
            txtsexoCF.Text = "FEMENINO";
        }
        string estadocivil = ObjCF.Rows[0][5].ToString();
        switch (estadocivil)
        {
            case "CA":
            {
                txtestadocivCF.Text = "CASADO";
                break;
            }
            case "SO":
            {
                txtestadocivCF.Text = "SOLTERO";
                break;
            }
            case "VI":
            {
                txtestadocivCF.Text = "VIUDO";
                break;
            }
            case "DI":
            {
                txtestadocivCF.Text = "DIVORCIO";
                break;
            }
        }

        txttelCasaCF.Text = ObjCF.Rows[0][6].ToString();
        txtteleTraCF.Text = ObjCF.Rows[0][7].ToString();
        txtteleCeluCF.Text = ObjCF.Rows[0][8].ToString();
        txtteleRECF.Text = ObjCF.Rows[0][9].ToString();
    }
    else
    {
        string script1 = @"<script type='text/javascript'>

```

```

                alert('El titular no puede carga familiar');
            </script>";
            ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script1, false);
        }

    }
    void fechaVencimiento()
    {
        lblnuevafecha.Text =
Convert.ToString(Convert.ToDateTime(txtfechaEmi.Text).AddMonths(3));
        if (Convert.ToDateTime(lblnuevafecha.Text) < DateTime.Now.Date)
        {
            lblestadoCop.Text = "VENCIDO";
        }
        else if (Convert.ToDateTime(lblnuevafecha.Text) > DateTime.Now.Date)
        {
            lblestadoCop.Text = "VIGENTE";
        }
    }

    protected void txtfechaEmi_TextChanged(object sender, EventArgs e)
    {
        if (DateTime.Parse(txtfechaEmi.Text) > DateTime.Now)
        {
            lblFecha.Visible = true;
            lblFecha.Text = "La fecha debe ser igual o inferior a la
actual";
            txtfechaEmi.Text = DateTime.Now.Date.ToString("d/MM/yyyy");
        }
        else
        {
            lblFecha.Visible = false;

            fechaVencimiento();
        }
    }
    protected void btnAgregarCaptado_Click(object sender, EventArgs e)
    {
        try
        {
            Captado cap = new Captado();
            CopiaLiteral CL = new CopiaLiteral();

            //INSERTANDO COPIA LITERAL
            CL.NumPartidaE1 = txtNumeroPartida.Text;
            CL.Direccion1 = txtDireccion.Text;
            CL.Manzana1 = txtManzana.Text;
            CL.Lote1 = txtLote.Text;
            CL.SubLote1 = txtSubLote.Text;
            CL.Referencia1 = txtReferencia.Text;
            CL.Numero1 = txtNumero.Text;
            CL.Interior1 = txtInterior.Text;
            if (cbdistrit.Text == "Seleccione Distrito..." || cbdistrit.Text
== "")
            {
                string script3 = @"<script type='text/javascript'>

```

```

        alert('SE NECESITA INGRESAR EL UBIGEO');
    </script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script3, false);
    }
    else
    {
        CL.IdDistrito1 = Convert.ToString(cbdistrit.Selectedvalue);
    }
    CL.AreaTerrenoTotal1 = txtAreaTotal.Text;
    CL.MedFrente1 = txtMedFrente.Text;
    CL.MedFondo1 = txtMedFondo.Text;
    CL.MedIzquierda1 = txtMedIzquierda.Text;
    CL.MedDerecha1 = txtMedDerecha.Text;
    CL.ObservacionCL1 = txtobsercopial.Text;
    CL.FechaEmision1 = Convert.ToDateTime(txtfechaEmi.Text);
//CAPTADO
    cap.IdPersona1 = lblpersona.Text;
    cap.CargaFamiliar1 = lblcargafamiliar.Text;
    if (dlltiparentesco.Text == "SELECCIONE")
    {
        string script2 = @"<script type='text/javascript'>
            alert('SELECCIONE EL PARENTESCO DE LA CARGA
FAMILIAR');
            </script>";
        ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script2, false);
    }
    else
    {
        cap.ParestescoCarga1 = dlltiparentesco.Text;
    }
    cap.EstadoCap1 = ddlescaptacion.Text;
    cap.Observacion1 = txtobservacion.Text;
    cap.UserReg1 = "US001";
//cap.UserReg1 = Variables.idusuario;
    cap.IpReg1 = "192.168.20.1";
    ConstructoraBol.Captado.insertar(cap, CL);
    string mensaje = "Ah registrado al captado";
    string script1 = @"<script type='text/javascript'>alert('" +
mensaje + "')</script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script1, false);
    }
        catch (Exception c)
        {
            string script1 = @"<script type='text/javascript'>
                alert('ERROR DE DATOS');
                </script>";
            ScriptManager.RegisterStartupScript(this,
typeof(Page), "alerta", script1, false);
        }
    }
}

```

```

BENEFICIARIO.aspx.cs
namespace Constructora.Sistema
{
    public partial class Beneficiario : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Variables.Pbenef != "A")
            {
                Response.Redirect("~/AccesoDenegado.aspx");
            }
            if (!IsPostBack)
            {
                llenar_Cmb(DdlEntidadTecnica);
                llenar_profesional(ddlProfesional);
            }
        }
        void llenar_Cmb(DropDownList cmb)
        {
            EntidadTecnica ET = new EntidadTecnica();
            cmb.DataSource = EntidadTecnica.LISTARENTIDADESTE();
            cmb.DataTextField = "RazonSocial";
            cmb.DataValueField = "IdEt";
            cmb.DataBind();
        }
        void cargargrillabene()
        {
            Captado obj = new Captado();
            obj.Dni1 = txtDni.Text;
            obj.UserReg1 = Variables.idusuario;
            dgvresultado.DataSource = Captado.buscarcapxdni(obj);
            dgvresultado.DataBind();
            btnGuardar.Enabled = true;
        }
        void llenar_profesional(DropDownList cmb)
        {
            cmb.DataSource = ProfesionalResidente.LLENARPROFERESI();
            cmb.DataTextField = "PROFESIONAL";
            cmb.DataValueField = "IdPr";
            cmb.DataBind();
        }
        protected void dgvresultado_SelectedIndexChanged(object sender,
        EventArgs e)
        {
            lblcaptadoid.Text = dgvresultado.SelectedRow.Cells[1].Text;
            txtnombreCompleto.Text = dgvresultado.SelectedRow.Cells[2].Text + "
            " + dgvresultado.SelectedRow.Cells[3].Text + " " +
            dgvresultado.SelectedRow.Cells[4].Text;
            lblestadocap.Text = dgvresultado.SelectedRow.Cells[8].Text;
            if (lblestadocap.Text == "BENEFICIARIO")
            {
                btnGuardar.Enabled = false;
            }
        }
        protected void btnGuardar_Click1(object sender, EventArgs e)
        {
            ConstructoraBol.Beneficiario bn = new
            ConstructoraBol.Beneficiario();
            bn.IdEt1 = Convert.ToString(DdlEntidadTecnica.SelectedValue);
            bn.IdCaptado1 = lblcaptadoid.Text;
            bn.IdEt1 = DdlEntidadTecnica.Text;
        }
    }
}

```

```

        bn.Modalidad1 = Ddlmodalidad.Text;
        bn.AreaConstruir1 = Convert.ToDecimal(txtareaconstruir.Text);
        bn.TazacionFabrica1 = Convert.ToDecimal(txttasacion.Text);
        bn.PresupuestoObra1 = Convert.ToDecimal(txtpresupuesto.Text);
        bn.ValorTerreno1 = Convert.ToDecimal(txtvalorterre.Text);
        bn.ValorBono1 = Convert.ToDecimal(txtvalorbono.Text);
        bn.Ahorro1 = Convert.ToDecimal(txtvalorahorro.Text);
        bn.TiempoEjecucion1 = Convert.ToInt16(txtempoejecu.Text);
        bn.OBSERVACION1 = txtObervacion.Text;
        bn.idUsuario1 = Variables.idusuario;
        bn.IdZona1 = Variables.idzona;
        bn.IpReg1 = "192.168.20.18";
        ConstructoraBol.Beneficiario.InsertarBene(bn);
        //MOSTRAR CONTRATO

        mostrarcontrato();
    }

        string script1 = @"<script type='text/javascript'>
            alert('EL CAPTADO YA ESTA INGRESADO COMO
BENEFICIARIO');
        </script>";
        ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script1, false);
        CleanControl(this.Controls);
    }

    }

    void mostrarcontrato()
    {
        ConstructoraBol.Beneficiario bene = new
ConstructoraBol.Beneficiario();
        bene.IdCaptado1 = lblcaptadoid.Text;
        lblNumContrato.Text
=ConstructoraBol.Beneficiario.Consultarcontrato(bene).Rows[0][0].ToString();

    }
    protected void btnBuscar_Click(object sender, EventArgs e)
    {
        cargagrillabene();
    }
}

```

```
APERTURAR CARTA FIANZA.aspx.cs

namespace Constructora.Sistema
{
    public partial class AperturarCF : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Variables.Pconst != "A")
            {
                Response.Redirect("~/AccesoDenegado.aspx");
            }

            if (!IsPostBack)
            {
                ConsularCFs();
            }
        }
        void ConsularCFs()
        {
            gvCF.DataSource = CartaFianza.consultaCF_Apertura();
            gvCF.DataBind();
        }
        void ConsultarCfFiltro()
        {
            CartaFianza obj = new CartaFianza();
            obj.CF1 = txtNombCF.Text;
            gvCF.DataSource = CartaFianza.consultaCF_AperturaFiltro(obj);
            gvCF.DataBind();
        }

        protected void btnMostrarTod_Click(object sender, EventArgs e)
        {
            ConsularCFs();
        }

        protected void btnBuscarCF_Click(object sender, EventArgs e)
        {
            ConsultarCfFiltro();
        }

        protected void gvCF_SelectedIndexChanged(object sender, EventArgs e)
        {
            lblIdConstru.Text= gvCF.SelectedRow.Cells[1].Text;
            lblIdCF.Text= gvCF.SelectedRow.Cells[2].Text;
            CartaFianza obj = new CartaFianza();
            obj.IdCF1 = lblIdCF.Text;
            DetConstruccion objC = new DetConstruccion();
            objC.IdConstru1 = lblIdConstru.Text;
            txtMinucipalidades.Text =
            CartaFianza.consultarMunicipalidadesCF(obj).Rows[0][0].ToString();
            gvBeneficiariosCF.DataSource =
            CartaFianza.consultarBeneficiariosCF(obj);
            gvBeneficiariosCF.DataBind();
        }
    }
}
```

```

    }

    protected void btnAperturarCF_Click(object sender, EventArgs e)
    {
        try
        {
            if (lblIdConstru.Text != "" && lblIdConstru.Text != "")
            {
                DetConstruccion objDetC = new DetConstruccion();
                CartaFianza objC = new CartaFianza();
                objC.IdCF1 = lblIdCF.Text;
                objDetC.IdConstru1 = lblIdConstru.Text;
                objDetC.User1 = Variables.idusuario;
                objDetC.Ip1 = "192.168.123.1";
                DetConstruccion.AperturarCF(objDetC, objC);
                string mensaje = "Ha aperturado la Carta Fianza : " +
gvCF.SelectedRow.Cells[3].Text;
                string script1 = @"<script type='text/javascript'>alert('" +
mensaje + "');</script>";
                ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script1, false);
                ConsularCFs();
            }
        }
        catch (Exception x)
        {
            string mensaje = "Ha ocurrido el siguiente error: " +
x.ToString();
            string script1 = @"<script type='text/javascript'>alert('" +
mensaje + "');</script>";
            ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script1, false);
        }
    }

    protected void gvCF_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if ((gvCF.Rows.Count < gvCF.PageSize) && (gvCF.Rows.Count +
gvCF.PageSize * gvCF.PageIndex < ((DataTable)(gvCF.DataSource)).Rows.Count))

            e.Row.Cells[1].Visible = false;
    }
}

```

## ActualizacionFMV.aspx.cs

```
namespace Constructora.Sistema
{
    public partial class actualizacionFMV : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Variables.Pbenef != "A")
            {
                Response.Redirect("~/AccesoDenegado.aspx");
            }
            if (!IsPostBack)
            {
                txtnombreCompleto0.Enabled = false;
            }
        }
        void cargargrillaCodigo()
        {
            ConstructoraBol.Beneficiario obj = new ConstructoraBol.Beneficiario();
            obj.Dni1 = lbldni.Text;
            obj.idUsuario1 = Variables.idusuario;
            dgvresultado.DataSource =
ConstructoraBol.Beneficiario.BuscarBeneCodigo(obj);
            dgvresultado.DataBind();
        }
        protected void Guardar_Click(object sender, EventArgs e)
        {
            try
            {
                ConstructoraBol.Beneficiario bene1 = new
ConstructoraBol.Beneficiario();
                bene1.IdBeneficiario1 = lblbeneficiario.Text;
                bene1.CodigoFmv1 = txtcodigo.Text;
                bene1.FechaCodFMV1 = Convert.ToDateTime(txtFecha.Text);
                bene1.UserCodi1 = Variables.idusuario;
                bene1.idUsuario1 = Variables.idusuario;
                bene1.IpReg1 = "192.168.20.10";
                bene1.MotivoReg1 = "INS.FCH.COD";
                bene1.Tipo1 = "I";
                bene1.Campo1 = "";
                bene1.Valor1 = "";

                ConstructoraBol.Beneficiario.ModificarCodigo(bene1);
                string script1 = @"<script type='text/javascript'>
alert('Ingreso satisfactorio.');"
</script>";
                ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script1, false);
                CleanControl(this.Controls);
                cargargrillaCodigo();
            }
            catch (Exception ex)
            {
                string script1 = @"<script type='text/javascript'>
alert('Error al insertar, verifique datos!');"
</script>";
            }
        }
    }
}
```

```

ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script1, false);

    }
}

protected void btnBuscar_Click(object sender, EventArgs e)
{
    //lbldni.Text = txtDni.Text;
    cargargrillaCodigo();
}

protected void dgvresultado_SelectedIndexChanged(object sender, EventArgs e)
{
    lblbeneficiario.Text = dgvresultado.SelectedRow.Cells[1].Text;
    txtnombreCompleto0.Text = dgvresultado.SelectedRow.Cells[2].Text + " " +
dgvresultado.SelectedRow.Cells[3].Text + " " +
dgvresultado.SelectedRow.Cells[4].Text;
    if (dgvresultado.SelectedRow.Cells[8].Text == "CODIGO")
    {
        txtcodigo.Enabled = false;
        txtFecha.Enabled = false;
        Guardar.Visible = false;
        txtcodigo.Text = dgvresultado.SelectedRow.Cells[11].Text;
        txtFecha.Text = dgvresultado.SelectedRow.Cells[12].Text;
        btnModificar.Visible = true;
    }
    else
    {
        txtcodigo.Enabled = true;
        txtFecha.Enabled = true;
        Guardar.Enabled = true;
    }
}

public void CleanControl(ControlCollection controles)
{
    foreach (Control control in controles)
    {
        if (control is TextBox)
            ((TextBox)control).Text = string.Empty;
        else if (control is DropDownList)
            ((DropDownList)control).ClearSelection();
        else if (control is RadioButtonList)
            ((RadioButtonList)control).ClearSelection();
        else if (control is CheckBoxList)
            ((CheckBoxList)control).ClearSelection();
        else if (control is RadioButton)
            ((RadioButton)control).Checked = false;
        else if (control is CheckBox)
            ((CheckBox)control).Checked = false;

        else if (control.HasControls())
            //Esta linea detecta un Control que contenga otros Controles
            //Así ningún control se quedará sin ser limpiado.
            CleanControl(control.Controls);
    }
}

```

```
protected void btnModificar_Click(object sender, EventArgs e)
{
    btncambioCodigo.Visible = true;
    btncambiofecha.Visible = true;
    txtcodigo.Enabled=true;
    txtFecha.Enabled=true;
    lblcodigo.Text = txtcodigo.Text;
    lblfecha.Text = txtFecha.Text;
}

protected void btncambioCodigo_Click(object sender, EventArgs e)
{
    try
    {
        if (txtcodigo.Text != lblcodigo.Text)
        {
            ConstructoraBol.Beneficiario bene1 = new
ConstructoraBol.Beneficiario();
            bene1.IdBeneficiario1 = lblbeneficiario.Text;
            bene1.CodigoFmv1 = txtcodigo.Text;
            bene1.FechaCodFMV1 = Convert.ToDateTime(lblfecha.Text);
            bene1.UserCod1 = Variables.idusuario;
            bene1.idUsuario1 = Variables.idusuario;
            bene1.IpReg1 = "192.168.20.10";
            bene1.MotivoReg1 = "MOD.NUM.COD";
            bene1.Tipo1 = "M";
            bene1.Campo1 = "CodigoFMV";
            bene1.Valor1 = txtcodigo.Text;

            ConstructoraBol.Beneficiario.ModificarCodigo(bene1);

            // CleanControl(this.Controls);
            cargagrillaCodigo();
            lblmensajecodigo.Text = "Modificación Exitosa de código";
        }
        else
        {
            lblmensajecodigo.Text = "No se realizo cambios en el código";
        }
    }
    catch (Exception ex)
    {
        string script1 = @"<script type='text/javascript'>
alert(¡Error al insertar, verifique datos!);
</script>";
ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script1, false);
    }
}

protected void btncambiofecha_Click(object sender, EventArgs e)
{
    try
    {
        if (txtFecha.Text != lblfecha.Text)
        {
```

```

        ConstructoraBol.Beneficiario bene1 = new
ConstructoraBol.Beneficiario();
        bene1.IdBeneficiario1 = lblbeneficiario.Text;
        bene1.CodigoFmv1 = lblcodigo.Text;
        bene1.FechaCodFMV1 = Convert.ToDateTime(txtFecha.Text);
        bene1.UserCodi1 = Variables.idusuario;
        bene1.idUsuario1 = Variables.idusuario;
        bene1.IpReg1 = "192.168.20.10";
        bene1.MotivoReg1 = "MOD.FCH.COD";
        bene1.Tipo1 = "M";
        bene1.Campo1 = "FechaCodigoFMV";
        bene1.Valor1 = txtFecha.Text;

        ConstructoraBol.Beneficiario.ModificarCodigo(bene1);
        //CleanControl(this.Controls);
        cargagrillaCodigo();
        lblmensajefecha.Text = "Modificación Exitosa de la fecha";
    }
    else
    {
        lblmensajefecha.Text = "No se realizo cambios en la fecha";
    }
}
catch (Exception ex)
{
    string script1 = @"<script type='text/javascript'>

        alert(¡Error al insertar, verifique datos!);
    </script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script1, false);
}
}

protected void Button1_Click1(object sender, EventArgs e)
{
    lbldni.Text = txtDni.Text;

    cargagrillaCodigo();
}

protected void txtFecha_TextChanged(object sender, EventArgs e)
{
    if (DateTime.Parse(txtFecha.Text) > DateTime.Now)
    {
        lblfechamens.Text = "La fecha debe ser igual o inferior a la actual";
        txtFecha.Text = DateTime.Now.Date.ToString("d/MM/yyyy");
    }
    else
        lblfechamens.Text = "";
}

```

## FECHA FIRMA CONTRATO.aspx.cs

```
namespace Constructora.Sistema
{
    public partial class ActualizarFechas : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Variables.Pbenef != "A")
            {
                Response.Redirect("~/AccesoDenegado.aspx");
            }
            if (!IsPostBack)
            {
            }
        }
        void cargagrillabene1()
        {
            ConstructoraBol.Beneficiario obj = new
ConstructoraBol.Beneficiario();
            obj.Dni1 = lbldni.Text;
            obj.IdZona1 = Variables.idzona;
            dgvresultado.DataSource =
ConstructoraBol.Beneficiario.buscarBENE(obj);
            dgvresultado.DataBind();
        }

        protected void dgvresultado_SelectedIndexChanged(object sender,
EventArgs e)
        {
            btnCambiosFirma.Visible = false;
            btnCambiosFMV.Visible = false;
            lblbenecod.Text = dgvresultado.SelectedRow.Cells[1].Text;
            txtnombreCompleto.Text = dgvresultado.SelectedRow.Cells[2].Text + " "
+ dgvresultado.SelectedRow.Cells[3].Text + " " +
dgvresultado.SelectedRow.Cells[4].Text;
            txtnumeroContrato.Text = dgvresultado.SelectedRow.Cells[7].Text;
            txtfirmacontrato.Text = dgvresultado.SelectedRow.Cells[10].Text;

            if (dgvresultado.SelectedRow.Cells[10].Text == "" ||
dgvresultado.SelectedRow.Cells[10].Text == "&nbsp;")
            {
                txtfirmacontrato.Text = "";
                txtfirmacontrato.Enabled = true;
                btnGuardarFirma.Visible = true;
            }
            else
            {
                txtfirmacontrato.Text = dgvresultado.SelectedRow.Cells[10].Text;
                txtfirmacontrato.Enabled = false;
                btnModificar.Visible = true;
                btnGuardarFirma.Visible = false;
            }
            if (dgvresultado.SelectedRow.Cells[11].Text == "" ||
dgvresultado.SelectedRow.Cells[11].Text == "&nbsp;")
            {
                txtfondo.Text = "";
                txtfondo.Enabled = true;
                btnGuardarFMV.Visible = true;
            }
        }
    }
}
```

```
    }
    else
    {
        txtfondo.Text = dgvresultado.SelectedRow.Cells[11].Text;
        txtfondo.Enabled = false;
        btnGuardarFMV.Visible = false;
    }
    txtnombreCompleto.Enabled = false;
    txtnumeroContrato.Enabled = false;
    if (txtfirmacontrato.Enabled==false || txtfondo.Enabled == false)
    {
        btnModificar.Visible = true;
    }
}

protected void btnBuscar_Click1(object sender, EventArgs e)
{
    cargagrillabene1();
}

protected void Button1_Click1(object sender, EventArgs e)
{
    lbldni.Text = txtDni.Text;
    cargagrillabene1();

    ////////////////

    btnModificar.Visible = false;
    //Button2.Visible = true;
    //btnGuardarModificar.Visible = false;
    lblmensaje.Text = "";
    //lblmensaje2.Text = "";
    txtfirmacontrato.Text = "";
    txtfirmacontrato.Enabled = true;
    txtfondo.Text = "";
    txtfondo.Enabled = true;
}

protected void txtfirmacontrato_TextChanged(object sender, EventArgs e)
{
}

protected void btnModificar_Click(object sender, EventArgs e)
{
    lblmensaje.Text = "";
    lblmensaje3.Text = "";
    if (txtfirmacontrato.Text != "")
    {
        txtfirmacontrato.Enabled = true;
        btnCambiosFirma.Visible = true;
    }
    if (txtfondo.Text != "")
    {
        txtfondo.Enabled = true;
        btnCambiosFMV.Visible = true;
    }
    lblvalorfirma.Text = txtfirmacontrato.Text;
    lblvalorFMV.Text = txtfondo.Text;
}
```

```

    }

    protected void btnGuardarFirma_Click(object sender, EventArgs e)
    {
        try
        {
            ConstructoraBol.Beneficiario bene = new
ConstructoraBol.Beneficiario();
            bene.IdBeneficiario1 = lblbenecod.Text;
            bene.FechaFirmaContrato1 =
Convert.ToDateTime(txtfirmacontrato.Text);
            bene.idUsuario1 = Variables.idusuario;
            bene.IpReg1 = "192.168.20.10";
            bene.MotivoReg1 = "INS.FCH.CONT";
            bene.Tipo1 = "I";
            bene.Campo1 = "";
            bene.Valor1 = "";

ConstructoraBol.Beneficiario.ModificarFechaContrato(bene);
            lblmensaje.Text = "Ingreso Satisfactorio de la Fecha de
Firma de contrato";

            txtfirmacontrato.Enabled = false;
            btnGuardarFirma.Visible = false;
            btnModificar.Visible = true;
            cargagrillabene1();
        }
        catch (Exception ex)
        {
            lblmensaje.Text = Convert.ToString(ex);
        }
    }

    protected void btnGuardarFMV_Click(object sender, EventArgs e)
    {
        try
        {
            ConstructoraBol.Beneficiario bene = new
ConstructoraBol.Beneficiario();
            bene.IdBeneficiario1 = lblbenecod.Text;
            bene.FechaRegistFMV1 = Convert.ToDateTime(txtfondo.Text);
            bene.idUsuario1 = Variables.idusuario;
            bene.IpReg1 = "192.168.20.10";
            bene.MotivoReg1 = "INS.FCH.FMV";
            bene.Tipo1 = "I";
            bene.Campo1 = "";
            bene.Valor1 = "";
            ConstructoraBol.Beneficiario.ModificarFondoMV(bene);
            lblmensaje3.Text = "Ingreso Satisfactorio al fondo Mi
vivienda";

            txtfondo.Enabled = false;
            btnGuardarFMV.Visible = false;
            btnModificar.Visible = true;
            cargagrillabene1();
        }
        catch (Exception ex)
        {
            lblmensaje3.Text = Convert.ToString(ex);
        }
    }

    protected void btnCambiosFirma_Click(object sender, EventArgs e)

```

```

    {
        try
        {
            if (lblvalorfirma.Text != txtfirmacontrato.Text)
            {
                ConstructoraBol.Beneficiario bene = new
ConstructoraBol.Beneficiario();
                bene.IdBeneficiario1 = lblbenecod.Text;
                bene.FechaFirmaContrato1 =
Convert.ToDateTime(txtfirmacontrato.Text);
                bene.idUsuario1 = Variables.idusuario;
                bene.IpReg1 = "192.168.20.10";
                bene.MotivoReg1 = "MOD.FCH.CONT";
                bene.Tipo1 = "M";
                bene.Campo1 = "FechaFirmaContrato";
                bene.Valor1 = lblvalorfirma.Text;
                ConstructoraBol.Beneficiario.ModificarFechaContrato(bene);
                lblmensaje.Text = "Modificacion exitosa de la Fecha de Firma
de contrato";

                txtfirmacontrato.Enabled = false;
                btnGuardarFirma.Visible = false;
                btnModificar.Visible = true;
                cargargrillabene1();
            }
            else
            {
                lblmensaje.Text = "No se realizo ninguna modificación";
            }
        }
        catch (Exception ex)
        {
            lblmensaje.Text = Convert.ToString(ex);
        }
    }

protected void btnCambiosFMV_Click(object sender, EventArgs e)
{
    try
    {
        if (lblvalorFMV.Text != txtfondo.Text)
        {
            ConstructoraBol.Beneficiario bene = new
ConstructoraBol.Beneficiario();
            bene.IdBeneficiario1 = lblbenecod.Text;
            bene.FechaRegistFMV1 = Convert.ToDateTime(txtfondo.Text);
            bene.idUsuario1 = Variables.idusuario;
            bene.IpReg1 = "192.168.20.10";
            bene.MotivoReg1 = "MOD.FCH.FMV";
            bene.Tipo1 = "M";
            bene.Campo1 = "FechaFMV";
            bene.Valor1 = lblvalorFMV.Text;
            ConstructoraBol.Beneficiario.ModificarFondoMV(bene);
            lblmensaje3.Text = "Modificacion exitosa al fondo Mi
vivienda";

            txtfondo.Enabled = false;
            btnGuardarFMV.Visible = false;
            btnModificar.Visible = true;
            cargargrillabene1();
        }
    }
}

```

```
        else
        {
            lblmensaje3.Text = "No se realizo ninguna Modificación";
        }
    }
    catch (Exception ex)
    {
        lblmensaje3.Text = Convert.ToString(ex);
    }
}
}
```